CLASSIFYING AND RESPONDING TO NETWORK INTRUSIONS

By

MARIA PAPADAKI

A thesis submitted to the University of Plymouth
in partial fulfilment for the degree of

DOCTOR OF PHILOSOPHY

School of Computing, Communications & Electronics
Faculty of Technology

June 2004

Classifying and Responding to Network Intrusions

Maria Papadaki
MSc

Intrusion detection systems (IDS) have been widely adopted within the IT community, as passive monitoring tools that report security related problems to system administrators. However, the increasing number and evolving complexity of attacks, along with the growth and complexity of networking infrastructures, has led to overwhelming numbers of IDS alerts, which allow significantly smaller timeframe for a human to respond. The need for automated response is therefore very much evident. However, the adoption of such approaches has been constrained by practical limitations and administrators' consequent mistrust of systems' abilities to issue appropriate responses.

The thesis presents a thorough analysis of the problem of intrusions, and identifies false alarms as the main obstacle to the adoption of automated response. A critical examination of existing automated response systems is provided, along with a discussion of why a new solution is needed. The thesis determines that, while the detection capabilities remain imperfect, the problem of false alarms cannot be eliminated. Automated response technology must take this into account, and instead focus upon avoiding the disruption of legitimate users and services in such scenarios. The overall aim of the research has therefore been to enhance the automated response process, by considering the context of an attack, and investigate and evaluate a means of making intelligent response decisions.

The realisation of this objective has included the formulation of a response-oriented taxonomy of intrusions, which is used as a basis to systematically study intrusions and understand the threats detected by an IDS. From this foundation, a novel Flexible Automated and Intelligent Responder (FAIR) architecture has been designed, as the basis from which flexible and escalating levels of response are offered, according to the context of an attack. The thesis describes the design and operation of the architecture, focusing upon the contextual factors influencing the response process, and the way they are measured and assessed to formulate response decisions. The architecture is underpinned by the use of response policies which provide a means to reflect the changing needs and characteristics of organisations.

The main concepts of the new architecture were validated via a proof-of-concept prototype system. A series of test scenarios were used to demonstrate how the context of an attack can influence the response decisions, and how the response policies can be customised and used to enable intelligent decisions. This helped to prove that the concept of flexible automated response is indeed viable, and that the research has provided a suitable contribution to knowledge in this important domain.

List of Contents

List of Figures

List of Tables

Acknowledgments

The work presented in this thesis represents the results of a three year investigation into various aspects of intrusion response, in the information security area. All work was undertaken within the Network Research Group, at University of Plymouth, Plymouth, United Kingdom.

I would principally like to acknowledge the contribution of the following people:

- Dr. Steven Furnell, my Director of Studies, whose knowledge, enthusiasm, skills, professionalism, and support provided the ideal basis for the work to carry on in the right direction, and for important research and professional skills to be acquired.

- Dr. Benn Lines, my Supervisor, whose help and support was available throughout the course of the research programme, in the form of regular supervision meetings, and who was another positive influence for my work.

- Prof. Paul Reynolds, my Supervisor, who also provided valuable advice, support and guidance throughout the research, especially during the final stages of the project, and the improvement of the thesis.

Thanks are also due to my colleagues at the Network Research Group, whose support and collaboration has been a great influence throughout the project. I would also like to acknowledge the contribution of S.Y. (Jim) Lee whose MSc project work complemented the investigation of attitudes towards automated response.

In addition, I would like to acknowledge the contribution of the Hellenic State Scholarships Foundation (S.S.F.), which provided the funding for my studies in the United Kingdom. Without their support, the completion of this degree would not have been possible.

Finally, I would like to thank my family, and my friends for offering plentiful support, and encouragement throughout the course of the last years.

Author's declaration

At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other University award.

This study was financed with the aid of a scholarship from the State Scholarships Foundation (S.S.F.) of Greece.

Relevant scientific seminars and conferences were regularly attended at which work was often presented; external institutions, international exhibitions were visited for consultation purposes and several papers prepared for publication, details of which are listed in the appendices.

Signed ....

Date ......4/10/04..............................

CHAPTER 1

INTRODUCTION

1 INTRODUCTION

The chapter provides an introduction to the context of this research, by providing an overview of the main issues associated with the subject of study. Then the aims and objectives of the research are established, followed by a brief summary of each chapter.

1.1 Detecting and Responding to Intrusions

The increasing level of attacks against IT systems represents an unavoidable reality of the Internet revolution. From the malicious activities of external hackers to deliberate misuse by organisational insiders, no sector has shown itself to be immune from attack; the provision of a public-facing server is effectively all that an organisation needs to do in order to establish itself as a potential target. Evidence of this problem is provided by the CERT Coordination Centre (CERT/CC), according to which the number of reported incidents in 2002 had increased by 3303%, as compared to the respective number in 1995 (2412 as opposed to 82,094 reported incidents in 1995 and 2002 respectively) (CERT/CC 2003). The associated financial losses from intrusions have also increased, and the CSI/FBI Computer Crime and Security Surveys have shown that the reported annual financial losses caused by security breaches in 2002 had been increased by 355% within the previous five years (Richardson 2003; Power 2002).

The growing dependence upon information technology and networked systems highlights the need for advanced security countermeasures to protect the newly formed computing and networking infrastructures. As a result, a number of security technologies and tools have been employed to combat the problem of attacks and protect system resources, examples of which are firewalls, virus scanners, cryptography, to name but a few. Each of these plays a distinct role and tries to address different aspects of security, however, security tools are not completely foolproof and despite the efforts to improve their effectiveness, attackers still manage to penetrate and hence compromise systems.

Having these scenarios in mind, Dorothy Denning introduced the concept of Intrusion Detection in 1987 (Denning 1987). According to her paper, when all the other safeguards fail, the Intrusion Detection System (IDS) will passively monitor network or system activity, looking for indications of security related problems. The concept of Intrusion Detection has become widely accepted since then, and numerous IDS tools have been developed in order to contribute to the goal for enhanced networks and systems security. In fact, the 2003 CSI/FBI survey (Richardson 2003) shows that over 70% of respondents have now adopted IDS products in their organisations.

Detecting intrusions is not a trivial undertaking, as research has proven over the past 16 years. So far the research community has faced many challenges in this area, resulting in the development of various detection technologies that can be employed to detect anomalies or misuse within systems and networks. As a result, research efforts have mainly focused upon the detection capability of systems. Meanwhile, the equally important issue of responding to the problem, has so far been given a lower priority by the research community. As a result, the majority of Intrusion Detection Systems currently rely upon human intervention by a system administrator in order to deal with a security problem. Thus, the level of automation in their response capability is limited to offering a variety of alerting and notification options in the form of console alerts, emails, pager, cellular phones or Simple Network Management Protocol (SNMP) traps (Bace and Mell 2001).

However, there are too many incidents happening too quickly for manual response to be sufficient, as networking infrastructures have become increasingly large and complicated, spanning over different sites, even countries, and accommodating thousands of users and heterogeneous systems. At the same time attackers' tools have become more sophisticated, allowing lower skilled attackers to perform complex attacks of distributed nature with minimal effort (Furnell 2001).

Further evidence of the problem is provided by the recent trend within the security community to incorporate automated response features into detection systems and thereby enable systems to respond in cases of suspicious attacks, by issuing actions such as resetting connections, blocking users and systems or terminating suspicious applications. However, this trend has not been greatly adopted by members of the information society (Messmer 2003a), as there are still several issues and challenges that need to be addressed if automated response is to prove successful.

The main problem with automated response lies on its adverse effects in case of a false alarm scenario, and specifically when the detection system mistakes legitimate activity for malicious. Given that the detection capability of IDS is not foolproof, the probability of such a scenario is quite realistic. If the detection system was limited to generating an alarm, the effect of the IDS would be nothing more serious than annoying the administrator with a lot of false alarms to deal with. Even if false alarms represent a significant problem, introducing a great deal of administrative overhead, and resulting in administrators ultimately ignoring the IDS, that situation could get worse. If automated response was enabled, then legitimate users and services could get disrupted, resulting in far more serious consequences. In another negative scenario, an attacker could use the automated responder for Denial of Service (DoS) attacks, if he manages to trick the system to believe there is an intrusion occurring. In such a case, legitimate users would be disrupted and business critical services interrupted, resulting in turning the response system into a DoS tool.

Hence, improving automated intrusion response represents an important area for research, allowing great scope for further improvement. The focus of this research has been the intrusion response decision process and specifically the contextual information that can enable an automated responder system to make informed decisions and eventually operate autonomously. The hypothesis behind this approach is that the more informed a decision is, the less likely it is to cause adverse effects in a false alarm scenario.

1.2 Aims and Objectives of the Research

This study is concerned with the issue of automated intrusion response and more specifically the identification of the main factors influencing the intrusion response process, enabling the design and evaluation of a novel architectural architecture for flexible intrusion response strategies.

The main objectives of the research can be summarised as follows.

1. Appreciate the problem of attacks in computing networks and examine the role of intrusion detection technologies as a security countermeasure.

2. Assess the current use of intrusion detection and response / prevention systems in IT environments and identify their weaknesses and requirements for improvement.

3. Systematically study intrusions, focusing upon detectable intrusions and suggesting general response mechanisms that could be suitable in different contexts.

4. Study the contextual factors that may influence intrusion response decisions in practice, and investigate how they can be assessed or measured within a system.

5. Examine how the above factors can influence the response decision process, and incorporate them into a wider architecture that would enhance the response capabilities of a system.

6. Validate the main concepts of the proposed architecture by means of a prototype implementation.

The objectives presented above correspond to the general sequence of the material presented in this thesis, the structure of which will be discussed in the next section.

The work carried out in this project forms part of ongoing research in relation to the Intrusion Monitoring System (IMS) architecture (Furnell 1995).

1.3 Thesis Structure

Chapter 2 focuses upon the issue of intrusions, by presenting the difficulties in securing IT systems, followed by the extent of the problems they have been causing in computing networks, since the Internet revolution. This is followed by an introduction to intrusion detection technologies, highlighting their role as a fallback option in the attempt to achieve increased security. The chapter then provides an overview of the Intrusion Detection concepts, and the main technologies used, aiming to highlight the significant challenges in their implementation.

Chapter 3 considers the issue of response, and begins with an introduction of the various forms of response, along with the actions that could be initiated as a result of an IDS alarm. It then continues with an overview of the response capabilities of current intrusion detection systems, highlighting their weaknesses in combating the problem. The chapter concludes with the case for the need of automated response, pointing to the requirement for further improvements.

Chapter 4 discusses the challenges in automating intrusion response and the various research efforts in the area. Initially, the attitudes of system administrators and security experts in general are examined, identifying the main shortcomings and limitations of current automated response technologies. The challenges in the enhancement of automated response are then further analysed, leading to the identification of research areas that could contribute to the problem. The chapter continues with a review of several research efforts in the area of automated intrusion response, identifying their strengths and limitations. After investigating the range of potential research areas, the scope and boundaries of this research are then determined.

Chapter 5 presents the first main contribution of this research, the Response-Oriented Taxonomy of Intrusions. Initially, the need for this new approach is justified, by reviewing current intrusion taxonomies, and highlighting their limitations in providing insight into the issue of intrusion response. The new taxonomy should provide insight into the process of selecting appropriate responses, and form the basis of decision-making in an automated responder system. Hence, the Response-Oriented Taxonomy of Intrusions is based upon the main intrusion characteristics that are relevant to the issue of response. The aim of this taxonomy is to consider incidents and identify their different results in different contexts. After the details of the taxonomy are presented, providing ratings for the results of the main intrusion categories, the chapter concludes with general observations about intrusions, and other response influencing factors, leading to the indication of generic response mechanisms, suitable for different types of intrusions, in different contexts.

Chapter 6 details the research by proceeding to identify the main factors involved in the response decision process, as an introduction to the concepts of intelligent response. The principal focus of the research is then presented, which is the conceptual architecture for a Flexible Automated Intelligent Responder system (entitled FAIR). Initially, the main modules of the FAIR system are described, followed by the presentation of its operational characteristics, as a novel approach to existing efforts in the domain of automated response. Focus of this research effort is the factors influencing intrusion response, and the architecture under which, they can be used, to enhance the response capabilities of a system. Thus, a detailed description of the contextual factors, which the FAIR system needs to assess in order to reach a decision, is provided. The description includes details about how these factors are assessed within a monitored system, whether they are static or dynamic, how they influence one another, and how they can be used in the response decision process. Finally, the Response Policy is presented, describing the response decision process in more detail.

Chapter 7 describes the implementation of a prototype system, which embodies a subset of the key elements of the proposed architecture, demonstrating its main concepts. It goes on to detail the aspects of the FAIR system that have been realised in practice. In addition, test scenarios are provided, in order to demonstrate how the Responder can adapt its decisions to reflect changes in the environment, including changes in the Responder and the targeted system. Finally, the user-friendly interface of the Response Policy Manager is described, highlighting its role in facilitating the process of customising Response Policies, and eventually enabling the reflection of experience from previous history of incidents within the Responder.

Finally, chapter 8 presents the main conclusions from this research, highlighting the principle achievements and limitations of the work, along with suggestions for potential further improvement.

The thesis also includes a number of appendices, which contain a variety of additional information in support of the main discussion (including a number of published papers from the research programme).

CHAPTER 2

INTRUSIONS AND INTRUSION DETECTION SYSTEMS

2 INTRUSIONS AND INTRUSION DETECTION SYSTEMS

This chapter begins by presenting the difficulties in securing systems, followed by the issue of intrusions and the extent of the problems they have been causing in computing networks, since the Internet revolution. An introduction of intrusion detection technologies follows, highlighting their role as a fallback option, when other security countermeasures fail to fulfil their goal. The chapter then provides an introduction to Intrusion Detection concepts, and an overview of the main technologies used.

2.1 Difficulties in achieving secure systems

Despite the fact that organisations are getting better at adopting security technologies, the problem of intrusions still remains a significant hurdle. This section considers the main barriers in achieving effective security countermeasures.

2.1.1 Protocol and software vulnerabilities

Initially, let us consider the problem of protocol and software vulnerabilities. There are three categories of flaws:

- Design Vulnerabilities. Flaws that could have been avoided at the design stage of a protocol or software. An example of such a flaw is at the ICMP Echo and Echo Reply messages (Ping) (Northcutt 1999), which are used to provide feedback about problems in network connectivity between hosts. Specifically, it is possible to verify if a host is 'alive' and the speed with which it can respond to your query. However, the protocol specification left out an important issue; the use of broadcast addresses. If an attacker pings a broadcast address with a spoofed IP address of a victim as source address, this will result in a denial of service attack, as all of the pinged machines will send a reply to the targeted system, resulting in flooding it with traffic (The incident described here is a smurf attack, the details of which can be found in the Common Vulnerabilities and Exposures (CVE) database, in the entry CVE-1999-0513).

- Implementation Vulnerabilities. Flaws that could be avoided at the implementation phase of a protocol or software. A very common example is buffer overflow vulnerabilities, which allow data to be stored in a buffer without ensuring firstly that the size of the data does not exceed the capacity of the buffer. Buffer overflow vulnerabilities can result in denial of service attacks or unauthorised access attempts (Aleph1 1996).

- Configuration Vulnerabilities. Flaws that could be avoided during the configuration of systems and networks. An example is the use of the default or null passwords that are provided by software vendors; they should be changed after the installation of the software. However, in many cases they are not, allowing intruders to use them and acquire unauthorised access to systems and networks. The SQLSnake / Spida worm, which was released in May 2002 (Dougherty 2002), exploits such a vulnerability, and specifically depends upon the SQL Server administrative account "*sa*" having the default null password, to cause denial of service, unauthorised access, and disclosure of information about the victim's system configuration. In fact, the SQLSnake / Spida worm was found to be the second most common vulnerability exploit in Windows systems during 2003 (SANS Institute 2003).

One would expect that commercial software would be supplied free from exploitable security vulnerabilities, that users would be aware of the potential security problems and would make the necessary efforts to configure their products according to their instructions; this is not always the case. For example, buffer overflow vulnerabilities have been known for some time, with the Morris worm in 1998 representing the first time that they were widely exploited on the Internet (Rogers 2001). Applying simple secure programming principles should be enough to avoid them, but they keep appearing in software products, offering endless opportunities to hackers (SANS 2003).

The problem is that from the software vendor's perspective, meeting deadlines and delivering products to the market is far more important than ensuring that the products are free from security problems. To date, it seems that bad publicity arising from vulnerabilities has not been a sufficient deterrent to stop people buying such software products, and the vendors have been content to address security problems later in the form of software upgrades or patches. Hence, the number of vulnerabilities keeps mounting, as the third Internet Threat Report indicates (Symantec 2003). Amongst other things, it states that there were over two thousand five hundred new vulnerabilities in 2002 (affecting more than two thousand products), an eighty one percent increase since the year before. In fact, according to the same report, there were seven new vulnerabilities documented, on average, for each day of the previous year.

Not surprisingly, the administrative overhead of making sure that all versions of software are updated regularly is a major challenge for system administrators. Hence, the findings by the Department of Trade and Industry, which state that businesses consider keeping up with vulnerabilities as their biggest problem with security (DTI 2002). In fact, the vast majority of the top twenty most exploited vulnerabilities (SANS 2003) are not new ones, as one would expect. In ninety percent of the cases, they have been known for more than a year (including the case of SQLSnake / Spida worm), and in many cases for more than three years. Yet, attackers can still count on the fact that they will find an abundance of vulnerable targets, and the fact that they can easily obtain attack tools for older and more widely known vulnerabilities makes their task even easier.

2.1.2 New risks associated with new technologies

As has been shown, the number of vulnerabilities associated with products keeps increasing, representing a significant task for administrators to keep up with them. However, the potential for products or technologies to introduce new risks is even more substantial when they are newly introduced. In fact, due to the unreliability of new technologies, many security-aware administrators do not install new applications within their systems until their first post-release patch has been issued. However, there are times when it simply is not possible to wait, as the advantages offered by new technologies are too attractive for organisations to resist; the convenience they offer, or the advantages they provide for competitive driven organisations, often outweigh the risks they introduce.

An example of such a case is the use of unregulated wireless networks, an increasingly popular technology for the flexibility it offers to overcome the limitations of traditional wired networking. At the same time, however, it makes networks vulnerable to establishing illegitimate connections, eavesdropping and achieving unauthorised access to legitimate users' resources, including software and data (Batista 2002). Since the commercial products, based on the Wireless LAN 802.11b standard, have weak encryption, which is not enabled by default, it is possible for an attacker to intercept radio waves of Wireless-LANs within a radius of up to a few hundred metres. In fact, when the network identifier and the administration password of the wireless access point are not changed from their default values, then there are even more opportunities for an attacker to abuse these networks. It is indicative that during the nine days of the 2002 World-Wide War Drive (WWWD 2002), an attempt to identify vulnerable wireless networks across the globe, around twenty five thousand wireless access points were identified, of which 72% did not have encryption enabled. Also, more than a third (35%) of them were still using the default network identifier. In this case, the problems associated with wireless technologies, are due to their limitations in design and implementation (weak encryption, authentication), and the general lack of awareness by users of how to deploy them (use of default configuration).

2.1.3 Ineffective use of security technologies

Another significant problem is the lack of defence in depth within organisations and the ineffective use of security technologies. While it has already been shown by the 8th CSI / FBI survey that organisations are becoming better at adopting security technologies, this does not mean that they are used effectively (Richardson 2003). A possible reason for this is given by the Global Information Security Survey, published by Ernst & Young (2003), which reveals that purchase of relevant technology is prioritised ahead of staff awareness and training, with 83% versus 29% of respondents listing them within their top three security spending areas. A consequence of the lack of attention to training is that, even though security technologies may have been purchased, they are not used correctly. For example, in the "E-security – 2002 and beyond" survey (3i 2002), it was revealed that around 80% of firewalls were inappropriately installed and / or maintained.

The problems of inadequate staff training and inappropriate risk assessment is also highlighted by the U.K. Audit Commission (2001), according to which these two problems were identified as the main reasons for being infected by viruses. In the same report, failures in basic security controls, such as appropriate supervision and effective access control, were found responsible for many cases of IT abuse.

In addition, the complexity of security tools does not make the problem any easier. In the same survey, complexity and interoperability issues of security tools have been identified as the main barrier to the growth of e-security. Unfortunately, there are too many firms offering products that address a particular problem, without thinking how their products interoperate with other security tools (current and future). Almost a quarter thought it was a problem then (2002) and almost half of respondents thought it would be a problem in 3 years time (3i 2002). At the same time, the complexity of managing so many different tools in one organisation, which do not often integrate with each other and require multiple internal skill-sets, makes the necessity for staff training more important and the task of securing systems even harder.

As a result, it is not surprising that the problem of attacks cannot be fully addressed with the purchase of security controls only. As long as there are organisations that are not aware of the risks they need to address, and employees not being trained about the risks associated with the technologies they use, there will always be potential for security breaches.

2.2 The Problem of Intrusions

A great enabler for the information revolution has been the Internet, and its ability to provide instant access to information and services from across the globe. However, as information and communications technologies have become ever more indispensable, the requirements for increased security in the Internet have had to change. In the effort to improve security, a number of security technologies have been deployed, with firewalls, encryption, user authentication, access control, anti-virus software and digital certificates being some typical examples. As the eighth annual computer crime and security survey reveals (Richardson 2003), an increasing number of organisations are now using them; virtually all organisations use anti-virus software and firewalls (99% and 98% respectively), while the vast majority use access control and physical security (92% and 91% respectively). Also, an increasing percentage of them are using encrypted files and digital certificates (69% and 49% respectively).

However, despite the fact that organisations are becoming better at adopting security technologies, the problem of attacks is still significant and many believe that it is still rising. The same survey (Richardson 2003) reported annual losses of about two hundred million US dollars, a decrease for the first time since 1999. However, the annual losses in 2003 were still significant and more than two times higher than the respective amount in 1997. According to CERT/CC (2003) on the other hand, there was an increasing number of security incidents reported this year; 114,855 incidents in the first three quarters of the year, as compared with 82,094 incidents for 2002. Figure 2.1 reflects the rising number of incidents reported to CERT/CC since the early 1990s.

Bearing in mind the statistics from CERT/CC, we can appreciate how significant the problem of intrusions is by considering the report from the U.K. Audit Commission (2001). According to their latest report, the average cost of one incident ranges from £6,000 to £36,000, according to the type of incident. Specifically, hacking incidents cause comparatively less damage, whereas the highest cost is caused by IT fraud incidents. As for malicious software, which represents the vast majority of IT incidents (Symantec 2003), the average cost of dealing with each incident reached £7,285 in 2000, more than three times higher than the respective amount (£1,700) in 1997. Finally, the 2002 Information Security Breaches Survey (DTI 2002), which targeted mostly small and medium enterprises, revealed that the average cost of a serious security incident was about £30,000, whereas in some cases the cost of individual incidents exceeded £500,000.

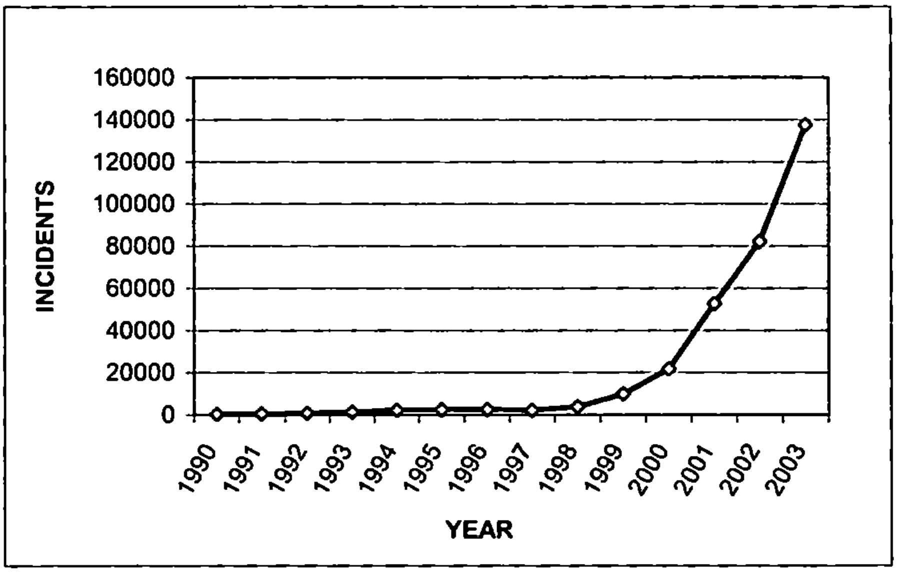


Figure 2.1 Number of incidents reported at CERT/CC

2.3 Principles of Intrusion Detection

In the previous sections it was shown that the barriers to achieving security are indeed difficult to overcome, leaving organisations vulnerable to the increasing problem of intrusions. It was shown that although organisations have become better at adopting security countermeasures, the problem of intrusions is still not eliminated, causing significant disturbance to organisations each year. Intrusion Detection technologies have been introduced with that fact in mind. Their role is to detect intrusions, when other countermeasures fail, by passively monitoring the events occurring in computer systems or networks and looking for security related problems. The intrusions they analyse are defined as attempts to compromise the confidentiality, integrity, availability, or to bypass the security mechanisms of a host or network (Bace and Mell 2001). There are several design approaches used for the development of Intrusion Detection Systems (IDS), mainly focusing upon their monitoring, analysis and response capabilities. This section provides an overview of the components that comprise an IDS and the approaches adopted for their development, aiming to offer a better understanding of the intrusion detection domain. An IDS can be described in terms of three fundamental logical components: sensors (or collectors), analysers, and user interface (Allen et. al. 2000).

2.3.1 Collection of Information – Sensors (or Collectors)

Sensors are responsible for collecting data and thus represent the information sources of an IDS. This information can be drawn from different sources, such as network packets, log files, and system calls traces. Sensors collect and forward this information to the analyser in order to determine whether an intrusion has taken place.

IDSs can be grouped into three categories, based upon the levels of a system at which they collect information (Bace and Mell 2001). These categories are Network, Host, Application, and are described in the sub-sections that follow, beginning with the lowest level of data collection. In each case, the information relating to advantages and disadvantages has been drawn from Allen et al (2000), and Bace and Mell (2001).

2.3.1.1 Network

Network-based IDSs represent the most popular option in the intrusion detection market, as most of the commercial systems employ this approach. A network-based IDS examines network traffic by using a set of single-purpose sensors or hosts placed at various points in a network. These units monitor network traffic, perform local analysis and report attacks or security-relevant events to a central management console. As the sensors are dedicated for the sole use of the IDS, they can be more easily secured against attacks and do not inflict any performance degradation on hosts. In fact, apart from being transparent to users, many of these sensors are designed to run in 'stealth' mode, in order to make it more difficult for an attacker to determine their presence and location.

The main advantages of network-based IDSs can be summarised as follows:

- A few well-placed network-based IDSs can monitor a large network and thus offer protection to a large number of hosts connected to it. Such a solution seems quite effective and thus attracts organisations that wish to offer the maximum protection possible at reasonably low cost.

- Network-based IDSs have minimal impact upon network infrastructures, since they listen on a network wire without interfering with the normal operation of a network. In the worst case, when they cannot handle the volume of traffic, it is their detection capability that will be compromised, not the network itself. Also, it is usually easy to deploy network-based IDS solutions with minimal effort.

- They can monitor a heterogeneous set of hosts and operating systems simultaneously, due to the fact that standard network protocols (e.g. TCP, UDP, IP) are supported and used by all networked systems.

- Network-based IDSs can be made very secure against attacks and even made invisible to many attackers.

The first two points in particular are generally considered to account for the popularity of the network-based approach amongst the majority of commercial intrusion detection systems. Although there is no conclusive proof for such a comment, since there are no market reports investigating the issue (Talisker 2001), an indication for its validity is the number of network-based commercial products available, against the host-based products. For example, in the Talisker's web site (Talisker 2003), which contains a list of intrusion detection products, there are 34 network based approaches, against 24 host based products.

However, the disadvantages of network-based IDSs are as follows:

- The processing load needed to monitor all packets in large or busy networks may impose significant overload upon the analysing engine of the IDS and, therefore, result in it failing to detect attacks above a specific amount of network traffic. Although some vendors have adopted hardware-based solutions for IDSs, in order to increase the speed of their processing capability, this simply serves to raise the threshold of traffic that can be handled (and the cost of implementation), while the inherent limitation still remains.

- The need to analyse packets as fast as possible, forces developers to look for fewer attacks and use as limited computing resources as possible. Thus detection effectiveness is often compromised for the sake of cost-effectiveness.

- Switched networks can limit the access of IDS sensors to subsets of network traffic. Switches subdivide networks into many small segments and provide dedicated links between hosts serviced by the same switch. At the same time, spanning ports, even if present, cannot solve the problem, as they cannot mirror all the traffic traversing the switch (Laing 2003).

- Network-based IDSs are limited to analysing only traffic in unencrypted form. However, encryption is becoming increasingly important, especially due to the emergence of Virtual Private Networks and IPSec (Bradbury 2003). Thus network-based IDSs would leave undetected attacks, such as Trojan horses, concealed in encrypted messages (Richard 2001).

- Some network-based IDSs have difficulty in dealing with network-based attacks that involve fragmented packets. As a result, the IDSs can become unstable and crash (Ptacek and Newsham 1998).

2.3.1.2 Host

Host-based IDSs monitor events occurring at a host level. They mainly use two types of information sources, namely operating system audit trails and system logs. Operating system audit trails are more detailed and better protected than system logs, since they are generated at the internal level (kernel) of an operating system. On the other hand, system logs are much simpler and smaller than audit trails, and hence can be easier to analyse. Some host-based IDSs are designed to report events to a centralized IDS management and reporting console, responsible for tracking many hosts, whereas others generate messages to report alerts to central network management systems.

The advantages of host based IDSs can be summarized into the following points:

- They can collect a vast amount of information from an individual computer system, allowing them to analyse events with greater reliability.

- They can detect attacks invisible to network-based IDSs, as they can monitor events local to a host and can access OS audit trails. For example, they can detect Trojan Horses or other software integrity breaches, by analysing OS audit trails and specifically by identifying inconsistencies in process execution.

- Host-based IDSs can also overcome the problem of encrypted environments, as information is usually in unencrypted form at the sending/receiving host.

- Host-based IDSs are unaffected by switched networks.

There are, however, disadvantages of host-based IDSs and these are summarized below:

- They are harder to configure and manage than network-based sensors, as information must be configured and managed for every host monitored. Given the great variety and number of hosts connected in large networks, this task can become even more demanding for cases of large organisations.

- If a host is targeted by an attack, then the sensor, or sometimes even the analysis engine, residing at the targeted host can become a victim of the attack. For instance, they can be disabled by certain denial-of-service attacks.

- Host-based IDSs are not suited for detecting network attacks that target an entire network or sub-network, such as scans, because the IDS can only see the network packets received by its host.

- The amount of information included in operating system audit trails can be immense, resulting in extensive requirements for local storage on a system. In general, they can inflict a performance degradation upon the monitored systems, as they consume computing resources of the hosts they are monitoring.

2.3.1.3 Application

An application-based IDS examines the behaviour of an application program by analysing events stored in its log files. The main role of application-based IDSs is to detect suspicious behaviour of authorised users exceeding their authorisation. An advantage of these IDSs is that they can overcome the problem of encrypted environments, since they interface with the application at transaction endpoints, where information is presented to users in unencrypted form.

However, application-based IDSs have a significant disadvantage, as the applications logs are not as well protected as other information sources, such as operating system audit trails. Also, application-based IDSs can often detect only a limited subset of attacks, usually focusing upon controlling user's permitted behaviour at the application level. Thus, application-based IDSs are usually integrated with host-based systems and used in combination with them.

2.3.2 Detection Models - Analysers

An Analyser receives input from one or more sensors, or from other analysers, and is responsible for determining if an intrusion has occurred. They can decide whether an intrusion is taking place or has already occurred, and provide evidence supporting that conclusion. In some cases, the Analyser will include suggestions on how to respond to the problem. Currently, there are two main approaches to determining the occurrence of intrusions: misuse detection and anomaly detection.

2.3.2.1 Misuse Detection

This approach is based upon the specification of what constitutes unauthorised behaviour and then the search for such behaviour within systems or networks to identify indications of known attacks. Since this model looks for patterns known to cause security problems, it is called a 'misuse' or 'attack signature' detection model (patterns corresponding to known attacks are called signatures).

The most traditional approach to misuse detection is the comparison of system or network activity to a predefined pattern of event(s) that describes a known attack. In this approach, each pattern of events specified corresponds to an attack as a separate signature. Newer and more sophisticated approaches, called "state-based" analysis techniques, can apply a single signature to detect groups of attacks, including variations of the same attack, enhancing in that way the detection capability of ID systems and limiting false alarms (Graham 2003).

A misuse detection model may be better suited for detecting known (or at least foreseen) intrusion techniques. For instance, repeated login attempts, malicious software, and exploits of other known software and protocol vulnerabilities can be detected by misuse signatures, enabling the system to specifically watch for signs of these occurrences. However, the definition of attack signatures, and especially in the most traditional approach, may not be comprehensive enough to ensure the coverage of all existing and future intrusion patterns, thus certain variation of intrusive behaviour may not be detected, resulting in a false negative case.

2.3.2.2 Anomaly Detection

The second approach is based upon the use of user, system or network profiles of normal behaviour, and searches for significant deviations from these profiles to detect security-related problems. It involves statistical analysis of parameters of a user's current session, system resources, or network traffic, which is used to determine whether these parameters exceed a certain threshold set by the specific model or by the security administrator. This detection approach is referred to as anomaly detection (Kumar and Spafford 1995). Such methods have traditionally been considered to be well suited for detecting masqueraders, since the activity observed from a masquerader is likely to deviate significantly from legitimate users' activity profiles (Mounji 1997). However, the concept of anomaly detection has lately been applied, in a wider context, to detect anomalies in the usage of network protocols, network and system resources as well (Debar 2003; Liston 2003).

2.3.2.3 Comparing Misuse with Anomaly Detection

The main advantages and disadvantages of misuse and anomaly detection can be summarised in Table 2.1.

Misuse Detection	Anomaly Detection
✔ Fewer false positive alarms	✗ More false positive alarms (have difficulty in determining whether unpredictable behaviour of users or elements of a system is indeed sufficiently abnormal to warrant concern)
✔ Can be immediately used for the protection of systems / networks	✗ Require training period (to characterise normal behaviour of users / system elements)
✗ Can only detect known attacks	✔ Can detect unknown attacks and variants of known attacks
✗ Must be constantly updated with signatures of new attacks	✔ Can detect cases of insider misuse, which involve abuse of user privileges, rather than exploit of security vulnerabilities (Allen et. al. 2000; Phyo and Furnell 2004)
✗ They often fail to detect variants of known attacks (state-based misuse detectors can overcome this limitation)	✔ The information they produce can be used to define signatures for misuse detectors

Table 2.1 Comparing Misuse - Anomaly Detection

It is important to point out that comparing misuse with anomaly detection does not aim to indicate which approach is more effective. Both approaches can complement one another, and at the same time, the more mechanisms are available for detecting an attack, the more effective an intrusion detection system can be. Despite the fact, though, that they can complement each other, the overall approach remains imperfect, with significant potential for false alarms. As such, the system cannot be completely certain in the correctness of its decisions, which in turn introduces the potential for doubt, and the need for flexibility, into any associated process of intrusion response.

2.3.3 User Interface

The User Interface enables an administrator to interact with the IDS by viewing security alerts about occurring or past intrusions or controlling its behaviour in terms of its detection and archiving features. In some systems with response capabilities, the administrator may be able to control the response behaviour of the system as well, by reviewing suggestions or authorising actions to be taken by the system. An Intrusion Detection System can interact with the following entities within a system:

- Systems Administrator

  The role of the administrator is to receive output from the IDS in the form of alerts, authorise the launch of specific actions in response to attacks, and modify the behaviour of the IDS (e.g. the degree of automation needed for the handling of attacks, the update of attack signatures, the sensitivity of sensors or the length of archived files).

- Network elements

  As a result of a response action being issued, an IDS might interact with various elements of a network, such as routers, firewalls, or network management systems.

- End users

  There are cases when interaction with the users is needed, as a result of the issue of a response action. Authentication challenge or limitation of access rights to the system might be an example of such actions. The operation of the IDS should ideally be as transparent to the users as possible, but such interaction is sometimes necessary in order to protect the system from reaching (or continue being in) a compromised state.

The great majority of the actions involved in the interaction of an IDS with other entities, represent actions issued in response to attacks and thus encompass the response capability of the IDS. Response actions are defined as the set of actions that the system takes once it detects intrusions or other events that appear suspicious from a security perspective. These are typically grouped into active and passive measures, with active measures involving some intervention on the part of the system, and passive measures involving reporting IDS findings to humans, who are then expected to take action based on those reports. The topic of response will be discussed in more detail in the next chapters.

2.4 Conclusions

This chapter initially considered the problem of intrusions, by including a discussion about the extent of the problem and the main obstacles organisations have to face to address it. It was shown that security is anything but a trivial process, with the problem of attacks rising, and causing significant damage to organisations worldwide, despite the increased efforts of organisations to employ security countermeasures and protect their systems.

Intrusion detection technologies have been introduced with that fact in mind; to look for indications of security related problems, when all the other security countermeasures fail to protect systems. The design approaches used for the development of Intrusion Detection Systems were then discussed, mainly focusing on their monitoring and analysis capabilities. For the monitoring capability, there were three main approaches, namely application, host and network-based IDSs, which collect security relevant events from various levels of system networks. A combination of all three approaches in a hybrid implementation is optimal, as it can offer the maximum protection at all levels of system functions. The analysis capability of IDSs can be characterised by the misuse and anomaly detection models; if used in combination with each other, they can enhance the detection capability of systems even further. However, even in that case, the detection capability of IDSs is not perfect, leading to many cases of false alarms.

The volume of potential intrusions means that security-conscious administrators have no choice but to use IDS if they want to stand a chance of managing the problem (Richardson 2003). Indeed, as the next two chapters will discuss, the volume of incidents is also likely to increase the desirability of automating the response process. However, the problem of false positive alarms stands as a main obstacle towards that direction. Thus it is important to appreciate that any attempt of incorporating intrusion response should take into account the problem of false positive alarms and the fact that detection cannot be 100% accurate.

CHAPTER 3

RESPONDING TO INTRUSIONS

3 RESPONDING TO INTRUSIONS

Having recognised the significance of intrusion detection and the issues associated with it, it is important to introduce the topic of intrusion response. This chapter begins by introducing intrusion response and the different forms that it might take. An overview of the response capabilities available in current intrusion detection systems is then presented, followed by a discussion about their strengths and limitations.

3.1 Forms of Response

Intrusion response is defined as the process of counteracting the effects of an intrusion. In the context of intrusion detection and response systems, it includes the series of actions taken by an Intrusion Detection System, following the detection of a security-related event. It is important to note that consideration is not only given to taking action after an intrusion has been detected, but also when events of interest take place and raise the level of alert in the system (i.e. during the early stages of a potential attack, when the system is suspecting the occurrence of an intrusion, but is not yet confident enough).

The aims of response actions, in general, can be summarised into the following (Furnell et. al. 2001):

1. Notify the administrator about the occurrence of an intrusion;

2. Collect more information about an incident;

3. Protect system resources;

- in the short term, this will include mechanisms to contain the intrusion, as well as to recover and restore the system to a well known state;

- in the long term, this may involve learning from the intrusion and using this knowledge to remove identified vulnerabilities in the system, and to enhance the detection and response capability. The underlying idea is to prevent reoccurrence of the intrusion;

4. Identify the perpetrator of the intrusion.

The process of response can either be conducted manually, automatically or as a combination of the two forms (Amoroso 1999). It is quite evident that automation can be more applicable for some cases of response actions than others. For example, actions aiming to notify an administrator about the occurrence of an intrusion can be more easily automated than identifying the perpetrator of the intrusion. The process of identifying the perpetrator often requires further investigation (also known as forensic analysis) (Honeynet Project 2000) and co-operation with other parties, such as Incident Response Teams and thus it naturally falls under the manually-oriented aspect of response. However, intrusion detection systems can assist this process by providing useful evidence about the incident, including the tracing of connections back to their entry point in the network.

The two main approaches to intrusion response are human/organisational and technical. The former are those that involve human processes and organisational structures, and may include actions such as reporting an incident to the police or invoking disciplinary procedures (e.g. in cases where internal personnel are responsible). By contrast, technical responses involve the use of functional techniques and software-based methods. These technical actions can themselves be further sub-classified, into either passive or active forms of response (Bace and Mell 2001).

3.1.1 Passive Response

Passive responses, as defined previously in Chapter 2, aim to notify other parties about the occurrence of an incident, relying upon them to take further actions. Passive actions may consist of:

1. Direct, location-dependent alerting options: display a pop up window on the IDS console screen, or generate SNMP traps to report to a central network management console.

2. Direct, location-independent alerting options: send alert via email, pager, web pages, Personal Digital Assistant (PDA), Short Message Service (SMS), etc.

3. Indirect alerting options: add an entry in a log file.

Passive responses, in the form of notifications and alerts, have traditionally been used since the conception of IDSs, primarily as an indicator of their detection effectiveness. Hence, they are still present in every intrusion detection product, offering the standard level of response. The fact that they have been tested for so long, and have been widely accepted, makes them the most common response option in commercial IDS systems to date.

3.1.2 Active Response

Active responses are the actions taken to counter the incident that has occurred. In a broad context, there is a wide range of actions that could be initiated for that purpose, including the approaches outlined in the paragraphs below. It should be noted that the responses are not mutually exclusive, and some may be issued in combination.

- Collecting more information about the incident.
    - Increasing monitoring level, by logging, for example, all the events generated in a suspicious session, or start monitoring the usage of system or network resources to ensure they are not abused.
    - Checking for the existence of vulnerabilities in the targeted systems.
    - Tracing connections back to their entry point in the network. This approach aims to identify the origins of offending traffic and provide evidence for further investigations regarding the incident (Wang et. al. 2001a).
    - Transparently authenticating users in the form of periodical or continuous keystroke analysis (Dowland et. al. 2002), or facial recognition (Lu et. al. 2003).
    - Issuing of explicit authentication challenges in the form of cognitive or associative questions (Irakleous et. al. 2002).

Ideally, actions aiming to collect more information about an incident should be non-intrusive to users or the attackers, given that they are normally used for cases of alarms with low possibility of corresponding to actual attacks. Thus, there is still a great chance, at that point, that the IDS is still wrong. However, in some cases it cannot be avoided. The inconvenience of asking users to re-authenticate themselves, for example, is less significant than risking the breach of their systems and thus the loss of their access altogether.

- Limiting permitted user behaviour.

  The aim of this approach is to ensure that the damage of an intrusion is minimised, by protecting important files, services or system resources. Thus, readjusting the rights of a potential impostor, who has been connected to a business-critical server to not allow access to the server software, will limit the ability of the attacker to create more damage. Another example could be the restriction of user rights to use shared devices, such as printers, or to perform unnecessary tasks, like installing new applications. The idea behind this approach is to limit the access of potential impostors to protect system resources, without actually disturbing their normal activity to any great degree.

- Blocking network traffic through firewalls and routers.

  Blocking actions involve the reconfiguration of firewalls or routers to block specific IP addresses or ports for a certain time period – for instance for an hour, a day, a week, or indefinitely. These actions are most applicable to attacks with high certainty, where there is no danger of disrupting legitimate users.

- Terminating network connections, user processes, or user sessions.

  Resetting network connections or disconnecting users are two more examples of actions that could be taken in response to attacks with high certainty, in cases where sustained access would put the system or user account at high risk.

- Saving content of unsaved work, or important information.

  These actions aim to minimise the effect of an attack, or a severe response, such as disconnecting users, by ensuring that no unsaved work or important information gets lost. In other words, it can be thought of as an emergency mini back-up process, triggered by the occurrence of an intrusion alarm, or the issue of a severe response.

- Using deception to limit the effect of an intrusion and guide the behaviour of the perpetrator to desired actions.

  Redirecting suspicious sessions to decoy systems is one example of deception, aiming to protect the originally attacked system and contain the attacker in a low-risk area. The difference between using this type of deception instead of disconnecting the attacker, for example, is that the attacker thinks the attack is successful and will not try another approach to achieve his goal. Decoy systems can be helpful in many other ways; for example, in a new virus outbreak, decoy systems could be made intentionally vulnerable to the virus and used as potential targets, to identify offending – infected hosts in the network and block them or disconnect them from the network. Warning the attacker about the alert status of the IDS, without actually revealing the source address of the IDS itself is another form of deception, which aims to discourage an attacker from carrying on with their plans. Introducing delay in suspicious network connections is also another example, aiming to discourage the attacker from carrying on, without actually revealing the presence of the IDS. Deception has been recently introduced as a form of response and although few research efforts have focused on it so far (Honeynet Project 2003; Cohen and Koike 2002), there is a great potential for this approach to enhance the response capabilities of IDSs.

Active responses can have a more significant impact upon a system, and thus they engender the danger of causing unwanted effects, in the event of them being falsely initiated. Not surprisingly, active responses have mainly been used in research prototype systems so far, and although there has recently been an increasing number of commercial systems utilising active response methods, (especially those that involve blocking of network traffic and termination of network connections), their application is still at an early stage and their effectiveness has not yet been proven conclusively. It should be noted, at this point, that more aggressive responses, which aim to launch attacks back at the attacker, have not been included in this study, due to their greater potential to cause damage. These responses, apart from being associated with several legal issues, they run a much greater risk of disrupting legitimate users in case of false alarm scenarios, or being utilised by attackers to indirectly launch attacks by turning the IDS into their accessory.

3.2 Intrusion Prevention Systems (IPS)

A recent trend in the intrusion detection and response domain is the emergence of intrusion prevention technologies. Although these incorporate intrusion detection mechanisms, they also have two significant differences. Instead of passively monitoring activity on systems and networks, they are positioned inline and can therefore block any unauthorised activity before it takes place (see Figure 3.1). In a network context, they can be thought of as sophisticated firewalls with intrusion detection capabilities, and in host environments, they monitor all system and API calls, blocking the ones that would result in malicious behaviour (Network Associates 2003d).

The biggest advantage of intrusion prevention is that it has the potential to respond in real time and stop attacks at the outset. However, as promising as it sounds, there are concerns about this approach. The first question that arises is the overhead they can introduce in networks and systems, by having to authorise all traffic and all occurring system calls. This overhead can become more significant in busy networks and servers, where performance is a crucial feature (Messmer 2002).

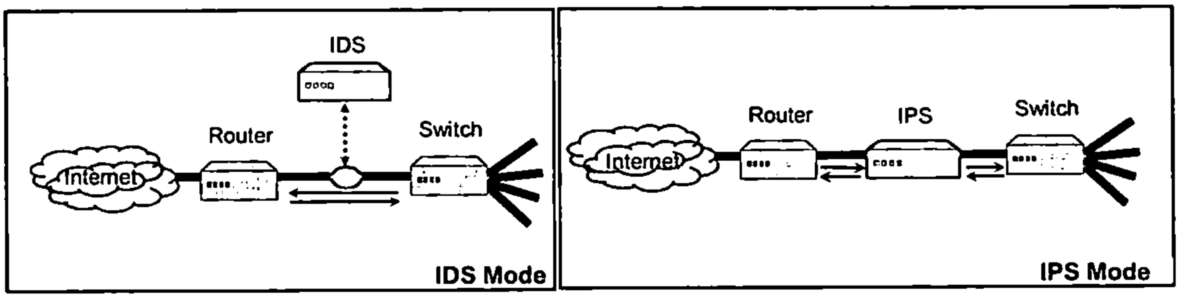


Figure 3.1 Offline vs. Inline IDS placement

Another potential problem is that they represent a single point of failure, which could have significant impact upon systems and networks. For example, in the scenario of an IPS crashing due to it being unable to handle the volume of network traffic, or being the target of an attack, the disturbance upon networks operation would be considerable. Although there are some efforts to overcome this problem (e.g. by using a back-up IPS that takes over in an emergency case; reconfiguring the router to redirect traffic around the IPS; or pre-configuring the IPS to run with minimum capabilities, allowing all traffic to pass), these solutions do not fully address the issue. The problem of handling large volumes of traffic has been already identified in intrusion detection systems, resulting in them crashing and being unable to operate, and although vendors are continuously improving their products, there are still issues to be addressed (Snyder 2003). Hence, intrusion prevention systems, which use the same detection mechanisms, have the potential to fall into the same category. However, since it is a relatively new technology, there have not been any reports or evaluation tests to confirm such a problem.

Another, and even more significant concern, is the problem of false alarms, and more importantly the false positive alarms. Although their ability to respond automatically to attacks and prevent them before they cause any damage is their major strength, IPSs always run the danger of denying access to legitimate users, or blocking legitimate traffic when they are mistaking it for malicious traffic. So far, intrusion prevention vendors have tried to address this problem by responding only to the attacks with high certainty (i.e. those that are not likely to create false positive alarms), and allowing all other system or network activity to pass (Network Associates 2003d). However, in this scenario, there is clearly potential for a whole range of other attacks with the potential to cause problems that do not get any response. Thus although the need for automating response and integrating intrusion detection with automated response is rising, there is still a great potential for further research, including the problem of addressing the uncertainty in intrusion response decisions.

3.3 A Survey of Prevention and Response Capabilities in Current IDS

This section presents a survey of the response capabilities within the leading commercial IDS products. The data was collected mainly from product literature, although in some cases, communication with vendors was necessary to extract more detailed information about some product features. The assessment was mainly focused upon the response options offered and, where possible, the response mechanism which initiates the responses.

Before presenting an overview of the response capabilities of IDSs, it is important to firstly refer to the changing nature of the domain, which has recently altered the attitudes of IDS vendors and the characteristics of the products they offer. The need for adoption of intrusion response methods has already been established with the emergence of Intrusion Prevention Systems. Although these solutions have been in use for several years, their adoption had been limited due to the concerns discussed in the previous section (Messmer 2003a). However, in June 2003 a market report published by Gartner (2003) had a significant impact upon the security community by suggesting the delay of large investments on intrusion detection systems, because they effectively add no practical value in enterprise security, and are going to be replaced by intrusion prevention products by 2005. The reaction to this report from IDS vendors and security specialists was very intense, leading to a long debate in the security community (Taylor and Wexler 2003).

The arguments against intrusion detection systems, presented in Gartner's report, were mainly focused on their inability to prevent intrusions and the vast amount of false positive alarms they generate. Although the last argument does represent a significant problem, the main reason for not having adopted automated response so far is the significant burden it introduces for system administrators. Given that intrusion prevention systems use the same detection methods as IDS, this argument could not work in favour of intrusion prevention systems either. However, the solutions IPS vendors provide do not actually address the problem of false positive alarms and thus in order to enable automated response, their products only block attacks, which are not prone to false alarms (attacks with high certainty). Although it could be argued that it is a sensible approach, and it can add another layer of security to systems, it does not mean that intrusion prevention products can replace IDS, as there will always be the need to detect all signs of intrusive behaviour, without the risk of overwhelming inline detection devices. Also, it should be possible to respond to a greater range of attacks, by not necessarily blocking them (if the certainty is not high enough), but by limiting the risk they pose, or at least collecting more information about them to enable a more certain judgement to be reached.

The effect of the debate against intrusion detection has had significant impact on the IDS market, forcing many vendors to incorporate intrusion prevention solutions in their products. Even if prevention solutions have not been adopted, it is noticeable that the term 'intrusion detection system' tends to be avoided and has been replaced with 'intrusion management system' or 'intrusion protection system', aiming to distance their products from any doubts that might be present in the minds of potential customers. Generally, it is fair to say that, although two years ago a limited number of IDSs used to offer active response methods, even if nearly all response actions were technically viable to implement (Lee 2001), automated response has now been adopted to a greater degree.

Finally, before presenting the response capabilities of current Intrusion Detection Systems, it is important to note that focus was given to the systems with more interesting approaches to intrusion response. The aim of the literature survey was to cover the most representative set of response options available, rather than reflect the degree to which automated active response has been adopted in the intrusion detection community. Also, putting aside the differences with IDS products, the response capabilities of IPSs have been included as well, since they do contribute to the understanding of the response domain and, in some cases, they have been integrated with intrusion detection solutions anyway. A summary of the products is provided at the end of the section, so a reader who is already familiar with them can refer to that section instead.

3.3.1 NFR Sentivist

NFR Sentivist is a network intrusion management system that provides by default a variety of notification options, in the form of console alerts, email and pager. Also, the user has the choice to enable notification via SNMP traps to IBM Tivoli and HP OpenView management systems. The user can additionally enable automated active response actions such as, resetting TCP connections and / or initiating firewall actions. Finally, it is possible to create alert responses that will automatically invoke third-party tools. For instance, such a tool would enable the tracing of suspicious connections.

The information provided for each alert includes type of attack, source and destination addresses, source and destination operating systems, raw packet capture data, help based on best practices, and a user-definable annotation field that aims to describe any user-defined response actions. (NFR Security 2003)

3.3.2 McAfee Intrushield

McAfee Intrushield is a network based approach that can be configured to detect and respond to attacks either passively as an IDS, or inline as an IPS. The range of response actions offered by Intrushield includes blocking of malicious packets, termination of connections, and reconfiguration of firewalls. In cases of successful attacks, Intrushield can log any subsequent communication between attacker and victim, aiming to enable forensics analysts to accurately assess the impact of an attack. Also, the alert notification options include alerts via console messages, email, pager and PDA. The user also has the option to use SNMP traps to forward alerts to central network management consoles, such as HP OpenView, IBM Tivoli or CA Unicenter. The information provided for each alert include fields such as the attack type, severity, and the source and destination addresses.

However, a very interesting feature of Intrushield is the Attack Verification facility, which can determine whether an attack has been successful, by utilizing stateful monitoring of traffic. According to this feature, each attack has the potential to be classified as failed, successful, blocked, suspicious or unknown (Network Associates 2003a). The *failed status* of an attack can be determined via stateful application protocol parsing. This enables the determination of whether the traffic in question constitutes a request to a server or a server response, and by recognising the different software implementations (e.g. Apache versus IIS web server), it is possible to determine if a request has been accepted or rejected by the server. Thus, if an attempt is against the wrong implementation or rejected by the server, the attack will have a failed status. The *success status* is more difficult to determine, as it depends upon the nature and aim of the attack. However, for cases of exploit-type attacks, success is achieved when a remote shell is obtained at the target. In IPS mode, when an attempt has been blocked, its status is characterised as *blocked*. Events related to suspicious activity have *suspicious status* and *unknown*, when there is not sufficient information to assess their result status. The importance of this feature lies in the fact that it has the potential to facilitate a learning process within a response system, by determining the outcome of an attack and thus the effectiveness of a response.

3.3.3 McAfee Entercept 4.0

Entercept is the host-based approach from McAfee that provides intrusion prevention solutions for desktops, web servers and database servers (Network Associates 2003b). The notification options it offers include email alerts, pager, SNMP traps, and spawning a process. Malicious activity is detected by intercepting system calls and prevented by blocking those that would result in malicious behaviour. Also, Entercept offers protection from Buffer Overflow attacks, with a patent-pending feature that prevents code execution as a result of a buffer overflow.

Entercept can offer scalable protection in the form of three security modes: Warning Mode, Protection Mode and Vault Mode. The Warning Mode is the least secure and is suitable for new servers, or the ones undergoing some change. In this mode, all malicious activity is logged, without being blocked. The Protection Mode employs misuse and anomaly detection to block known and unknown attacks, offering at the same time protection for buffer overflow attacks. Also, it is not possible for a user to elevate the privileges specified for their account, preventing in this way exploits for obtaining root-access privileges. The most secure option is the Vault Mode, which is designed for critical servers that do not need to change and need to be locked down. Specifically, Entercept prevents all the behaviour described already, plus any access to critical files of the operating system by users, even administrators. In this way the integrity of the operating system is safeguarded, as it is not possible to install rootkits, or Trojan versions of systems files (Network Associates 2003c).

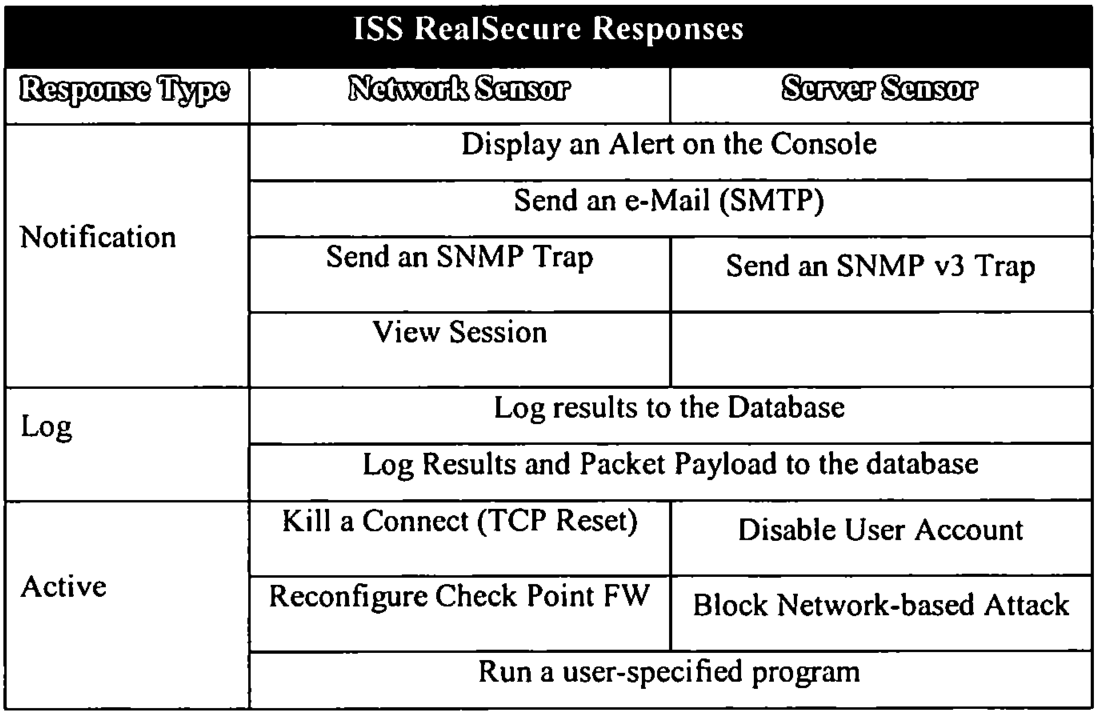
3.3.4 Symantec / Axent NetProwler

NetProwler is a network-based IDS, originally released by Axent Technologies. If an attack takes place, NetProwler reacts in real-time to protect the network, by terminating the offending traffic and/or modifying the firewall policy. NetProwler is armed with capabilities such as session logging, termination, capture, reporting, alerting, and firewall hardening (Symantec 2001). It also includes a variety of notification options, such as console alert, SNMP traps, email, pager, and HTML report forwarding. Symantec has recently acquired Axent Technologies, and has ended technical support services for the product in October 2003. Even so, NetProwler has played an important role in the IDS market, and the reason it is included in the survey is that it represents the baseline of active response products, in the sense that it only supports the most basic responses, such as TCP resets, firewall configuration, and collection of forensic evidence.

3.3.5 Symantec Manhunt

Manhunt is a network based approach that incorporates forensic and connection tracing capabilities. It can record traffic related to an intrusion, so that it can be used for forensic analysis and it can trace connections back to their entry in the network. In fact, the tracing analysis could be extended to other administrative domains, as long as they use Manhunt sensors as well. Other response options include session termination and notification in the form of email, SNMP traps and console alerts (Symantec 2003b).

3.3.6 ISS Proventia - Site Protector



ISS RealSecure Responses		
Response Type	Network Sensor	Server Sensor
Notification	Display an Alert on the Console	
	Send an e-Mail (SMTP)	
	Send an SNMP Trap	Send an SNMP v3 Trap
	View Session	
Log	Log results to the Database	
	Log Results and Packet Payload to the database	
Active	Kill a Connect (TCP Reset)	Disable User Account
	Reconfigure Check Point FW	Block Network-based Attack
	Run a user-specified program	

Table 3.1 ISS RealSecure Responses

Proventia is a new solution offered by Internet Security Systems, incorporating intrusion detection, intrusion prevention, firewalls, VPN and anti-virus technologies in one device. When detecting an attack, Proventia devices respond by logging events, packets or other evidence for forensic purposes. Also it is possible to reset TCP connections, and reconfigure firewalls or routers to block specific traffic. The alerting options provided are the standard ones (console alerts, email, pager, and SNMP traps) (Internet Security Systems 2003). Proventia is the newest member of the ISS unified management system, which includes network, server and desktop protection agents, all centrally managed by the Site Protector. The response capabilities of the ISS detection sensors are summarised in Table 3.1 (Internet Security Systems 2003b). As a note, the option "Execute a user-specified program" includes responses in executable binary form (or batch file/shell script); initiating a pager call, playing a sound, or reconfiguring a network device that does not have an API for management are examples of such actions.

An important module of Site Protector is the Security Fusion Module, which is responsible for identifying the impact of incoming intrusions, aiming to reduce the number of false alarms. Thus the impact of attacks can be classified in more meaningful ways, according to whether they target vulnerable systems, or whether a blocking action has been issued by the IDS sensors already (see Table 3.2). Also, the Security Fusion Module monitors attack activity over extended periods of time looking for patterns of activity, such as worms or targeted probes (Internet Security Systems 2003c).

Attack Status	Conditions
Success likely	Target vulnerable
Unknown impact	Not scanned recently
	Fusion not enabled
	Vulnerability check indeterminate
	Operating system check indeterminate
	No correlation
Failure likely	Blocked some packets
	No vulnerability
	Wrong operating system
Failed attack	Blocked at host
	Confirmed by sensor

Table 3.2 ISS Security Fusion Module: Attack Status Names

3.3.7 Cisco IDS – Intelligent Threat Investigation

Cisco's intrusion detection solution includes host and network modules. The Host module, Cisco Security Agent, has intrusion prevention features, and sits between the application level and the kernel, making instant decisions on whether to authorise or deny system activity. The user has the option to disable the protection features and run the agent in 'IDS mode', where activity is not blocked, but alerted (Cisco 2003). The alerts are initially sent to a central management console, a Director, which in turn uses notifications in the form of email or pager to alert the system administrator (Cisco 2003d). The network module uses intrusion prevention features as well, including dropping the packet, terminating a session, and reconfiguring routers, switches, or firewalls to shun specific IP addresses or ports. (Cisco 2003b)

Cisco Threat Response is a newly introduced feature that is not available for sale at the time of writing (there is a free trial version), but is going to be offered in the near future. It works with Cisco IDS sensors and aims to determine whether an attack was successful, by following a three-phase approach. Initially it performs a *basic investigation of a target vulnerability*, by examining the operating system version, patch levels, web services (when applicable) of the target, in order to determine if the attack has potential to succeed. For attacks with potential for success, an *advanced investigation of target* system logs, web logs, or other relevant data is performed, to determine if the attack has indeed succeeded. In cases of verified threats, the system initiates a *forensic data capture* phase, which includes safe storage of audit trails, log files, and intrusion traces from the targeted system. (Cisco 2003c)

3.3.8 Summary

The passive response capabilities of current commercial IDSs has been summarised in Table 3.3, in which notifications in the form of console alerts, email, pager or SNMP traps are clearly commonplace amongst the products. In some cases, notification via PDA, or forwarding of alert reports in HTML form is supported.

Passive Responses							
IDS name	NB / HB*	Console Alert	Email	Pager	SNMP	Html	PDA
NFR Sentivist	NB	✓	✓	✓	✓		
McAfee Intrushield	NB	✓	✓	✓	✓		✓
McAfee Entercept 4.0	HB		✓	✓	✓		
Symantec NetProwler	NB	✓	✓	✓	✓	✓	
Symantec Manhunt	NB	✓	✓		✓		
ISS RealSecure - Proventia	HB / NB	✓	✓	✓	✓		
Cisco IDS	HB / NB	✓	✓	✓			

*** NB / HB: Network Based / Host Based**

Table 3.3 Passive response capabilities of IDSs

The active response and prevention features of intrusion detection products has been summarised in Table 3.4, in which the most common response options for network-based systems is the resetting of TCP connections, and blocking network traffic by reconfiguring Access Control Lists of firewalls, routers or switches. In fact, nearly all network-based IDSs offer these options as a common means of defence. Tracing connections is not very common, however, collecting forensic evidence for security incidents is increasingly popular.

Active Responses										
IDS name	NB / HB*	Reset TCP Conn.	Block N/W traffic	Trace Conn.	Attack Verification	Forensic evidence	Block system call / process	Block User	Lock O/S	Enable third-party tools
NFR Sentivist	NB	✓	✓							✓
McAfee Intrushield	NB	✓	✓		✓	✓				
Symantec NetProwler	NB	✓	✓			✓				
Symantec Manhunt	NB	✓		✓		✓				
McAfee Entercept 4.0	HB						✓		✓	
ISS RealSecure ◦ Proventia	HB / NB	✓	✓		✓	✓		✓		✓
Cisco IDS	HB / NB	✓	✓		✓	✓	✓			

*** NB / HB: Network Based / Host Based**

Table 3.4 Active Response and Prevention capabilities of IDSs

For host-based approaches, the adoption of response and prevention actions varies more, resulting in the inability to state which ones are the most common. Providing forensic evidence and blocking system calls or processes seem to be the most popular options.

Blocking a user by disabling an account is another option, whereas locking the operating system, by preventing any changes to it, is a more severe approach suggested for critical servers that do not need to change.

3.4 Conclusions

This chapter introduced the issue of intrusion response and the various forms that it might take. Focus was given on the response options offered by intrusion detection systems, which can be generally characterised as passive or active response methods. The more traditional methods are passive, and aim to only notify other parties about the occurrence of an attack, relying upon them to take further actions about it. On the other hand, active responses offer a greater level of protection since they include actions that can actually counter attacks, but they represent a relatively new area in the intrusion detection domain and thus there is still potential for further improvement.

After reviewing the response and prevention capabilities of intrusion detection products it can be noted as a general remark that there is a relatively limited range of actions used for active response and prevention, suitable only for attacks with high certainty. There seems to be a lack of consideration for all the other cases, when the probability of mistaking normal activity for an intrusion is still considerable and when relatively less severe actions could usefully limit the damage of the suspected intrusion, without removing the danger altogether. A more flexible approach that would offer various escalating levels of response according to several contextual factors would enhance the response capability of intrusion detection systems even further. Thus, there is potential for further research in limiting the uncertainty in response decisions and improve the problem caused by the adverse effects of automated response in case of false positive alarms.

However, there are also some encouraging signs for the future of response-enabled IDSs, as some IDS vendors have incorporated more active features into their products. However, one aspect that has not been well portrayed in the literature review is the degree to which such responses are automated and enabled by default, as opposed to merely being provided as an option for any administrators willing to use them. The degree to which automated active response and prevention options are actually used is therefore worthy of further investigation, and represents one of the themes explored within the next chapter.

CHAPTER 4

AUTOMATING INTRUSION RESPONSE

4 AUTOMATING INTRUSION RESPONSE

This chapter considers the issue of automated response and the problems associated with it. Initially, the importance of automated response is discussed, followed by an identification of the difficulties and problems that prevent its adoption in the IT community. To reinforce the reluctance of systems and network administrators to use automated response, even if it is offered as an option by IDS vendors, a survey reflecting security specialists' views is then presented. Finally, current research efforts in the area of automated intrusion response are reviewed, aiming to provide a better understanding of the area and the challenges that are being addressed.

4.1 The need for automation

Adopting automated response is not only a desirable feature that would enhance the defence of networked systems against attacks; in many cases it is a necessity in order to keep up with the speed of evolving threats and provide a sufficient level of protection. Generally, and regardless of the recent trends in the area of intrusion response and prevention, the detection of a suspected intrusion would typically trigger a manual intervention by a system administrator, after having received an alert message from the intrusion detection system. The IDS would additionally assist the incident response process, by providing the details of the attack, saved in a log file (Bace and Mell 2001). However, responding manually to intrusions is not an easy task, as it can represent a significant administrative overhead that may involve dealing with a high number of alerts and notifications from the IDS, ensuring awareness of security bulletins and advisories from incident response teams, and taking appropriate actions to resolve each of the alerts reported. The number of alerts, and the respective administrative problem that they represent, would become even bigger in cases of the increasingly large and complicated IT infrastructures, which expand to different geographical sites and support activity of thousands of users. From the system administrator's perspective, the main requirement is to ensure that the system remains operational and available – this is what the users expect, and complaints will quickly occur if this is not the case. So, unless resolving a reported incident is explicitly required to ensure that this is the case, then the task is likely to be given a lower priority. The problem of addressing software vulnerabilities has already been identified in Chapter 2, where it was shown that although the vast majority of worms and other attacks are usually exploiting only a small number of vulnerabilities from operating system services, there is still an abundance of vulnerable systems. In fact, it was also shown that keeping up with vulnerabilities has been perceived by organisations to be their major problem with security. Thus, if addressing vulnerabilities introduces such a significant overhead for administrators, dealing manually with a great volume of IDS alarms would have even less chances of succeeding as an option.

Another limitation of manual response lies in the fact that it will often only be available during work hours, leaving a large window of opportunity for attackers to act undisturbed for the rest of the time. Bearing in mind that attacks can originate from across the globe, from different time zones, at any time of day, the effectiveness of manual response diminishes even further. According to Symantec's Internet Threat Report III, the attack activity tends to peak globally between 1:00 p.m. GMT and 10:00 p.m. GMT, regardless of the geographic location of the target (Symantec 2003). So, an organisation in China would be subjected to the majority of attacks during 10 p.m. and 7 a.m., when monitoring staff may be unavailable to respond. The unavailability of manual response could be an even more significant problem during weekends or holiday periods. In the same report from

Symantec, it was revealed that although the attack activity declines during weekends, there is still a considerable percentage (about 20%) of attackers who stay active during that time.

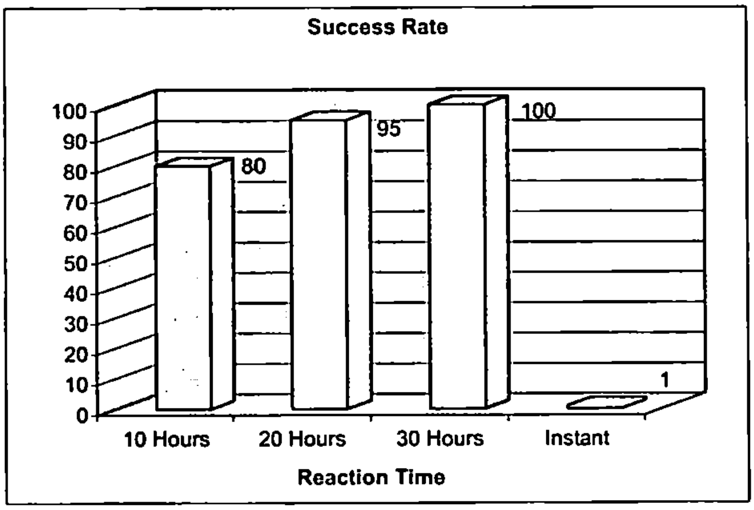
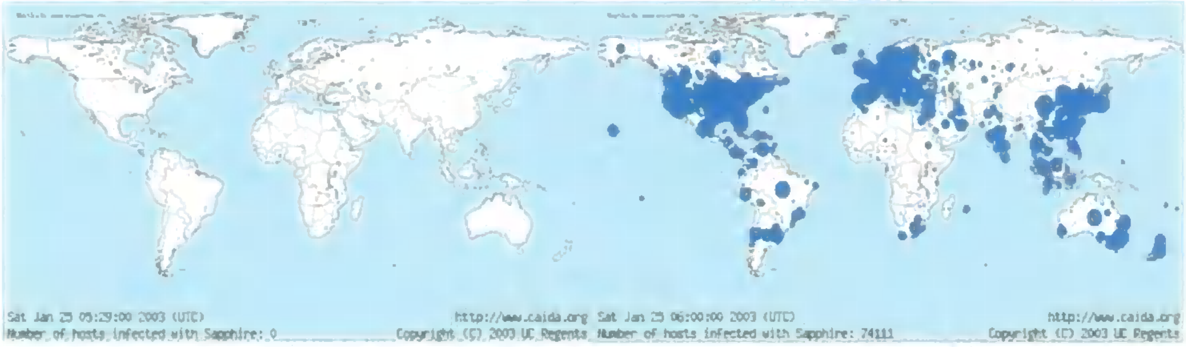


Figure 4.1 Attack Success Rate, according to the Reaction Time

The effect of reaction time on the success rate of attacks was demonstrated by Cohen, who carried out a simulation of attacks, defences and their consequences in complex cyber systems (Cohen 1999; Carver et al. 2001). The results indicate that if skilled attackers are given 10 hours after they are detected before a response, they will be successful 80% of the time. If they are given 20 hours, they will succeed 95% of the time. At 30 hours, the attacker almost never fails. The results also indicate that if a skilled attacker is given more than 30 hours, the skill of the system administrator will make no difference, as the attacker will succeed irrespectively. On the other hand, if the response is instant, the probability of a successful attack against a skilled system administrator is almost zero (see Figure 4.1). This strongly suggests that there is a relationship between the effectiveness of response and the time it is issued, and that there is a greater window of opportunity for an attacker if response is not issued on time.

In addition, manual response requires storage of large volumes of alarms for long periods of time, in order to allow an administrator to review them. The volume of alarms and the significant requirements for storage they pose was verified in the recent testing of IDS products by Snyder (2003), which observed that normal daily activity can generate as many as 100,000 alerts for viruses alone, and consequently cause significant storage and processing overheads. In fact, the volume of alarms and the storage required can be so significant, that according to Newman et al. (2002), there are cases of IDS products configured, by default, to retain locally only one day old alarms, making it very difficult for administrators to recover alerts that occurred over a weekend or previous days. Thus, it is not only a matter of increased storage requirements, but also a matter of preventing security events being ignored, or discarded, due to the time they were initiated.

Apart from the problems caused by the large volume of IDS alarms, another significant limitation of manual response relates to the change of attackers techniques, including the widespread use of automated scripts to generate attacks of distributed nature (Cheung and Levitt 1997). Here, the ability to respond manually is diminished even further, as the available time to do so is practically none. An example of such a case was the Slammer (or Sapphire) worm, which was initially released on 25 January 2003 and rapidly managed to disrupt the normal operation of systems worldwide, by generating a large volume of traffic and significantly degrading the overall network performance within a few hours (Roculan 2003). According to computer scientists at the University of California, and Eureka-based Silicon Defense, the Sapphire worm doubled its numbers every 8.5 seconds during the first minute of its attack and had managed to infect about 75,000 vulnerable systems within the first 10 minutes of its debut. The infection rate of Sapphire is reflected in Figure 4.2, which depicts the number of infected hosts within the first half hour from its debut. It was characterised as the fastest computer worm ever recorded (Moore et al. 2003), and it is notable that its infection rate was eventually slowed down only by the insufficiency of bandwidth that could not accommodate its growth (Moore et al. 2003b). The speed of manual response is not adequate in such cases, and thus the capability of a system to automatically respond and encounter intrusions is of increased importance.


Sat Jan 25 05:29:00 2003 (UTC)
http://www.caida.org
Number of hosts infected with Sapphire: 0
Copyright (C) 2003 UC Regents
Sat Jan 25 06:00:00 2003 (UTC)
http://www.caida.org
Number of hosts infected with Sapphire: 74111
Copyright (C) 2003 UC Regents

Source: http://www.caida.org/

Figure 4.2 Number of hosts infected with Sapphire (30 minutes after its debut)

4.2 Problems with automated response

Unfortunately, applying automated response is not such a straightforward solution, as there are dangers associated with it, if not used carefully. This is evidenced by the fact that while passive responses have already been automated to a large extent, most active responses remain manually initiated. While passive response actions will have little impact upon a system if they are initiated in a false positive alarm scenario, active responses could cause disruption to legitimate users, affect their access level to the system or even cause an unintentional denial of service attack to the system itself. According to Ptacek and Newsham (1998) a reactive intrusion detection system can be abused by an attacker where they do not require any response from the receiving host, for example, in cases of attacks over connectionless protocols. In such case, the attacker can blindly send forged packets, making the IDS think that the attack is coming from somewhere else, and resulting in the

IDS blocking the wrong system. However, even in cases of attacks over connection-oriented protocols, such as TCP, where the IDS does not identify the three-way handshake correctly, an attacker could as easily fake attacks, resulting in the IDS resetting legitimate TCP connections. Examples of such attacks are ICMP ping floods, SYN Flood, "Ping of Death" attack, or UDP packet storms. If, for example, the 'real' target of the attack (the receiver of the IDS response) were the network's DNS servers, then the impact of the attack would affect the entire network with much more serious consequences. The advantage, from the attacker's perspective, would be that he would not really need to generate vast amounts of traffic to overwhelm the DNS server himself, as the IDS would block it for a longer period of time, without the attacker needing to put so much effort. Hence it is important to make sure that when active response actions are launched automatically - without prior human authorisation - they have the predicted effects and do not put the system at a greater risk than it already is at the time. This situation requires confidence in the detection as well as the response capability of a system, which are not easy to achieve.

The detection capability of IDS systems has not improved dramatically over the last two years, as Snyder (2003) has noted in his test of IDS products. As for the response aspect, it is fair to say that it is still in its infancy. It is indicative to say that only 15% of the IDS systems surveyed by Axelsson (2000) were offering active responses at that time. A similar conclusion was drawn by Lee (2001), where a very limited number of response methods were initiated automatically in IDSs at the time. Nonetheless, after a study into the potential of response methods to become automated, Lee estimated that 33% of manual response actions could safely become automated, without having to further enhance the detection or response capability of the intrusion detection systems. Recently, as discussed in the previous chapter, the adoption of active responses has increased, and intrusion detection systems have started offering response and prevention options to a set of attacks with high certainty. However, their level of automation is not always high, as in many cases the automation of active responses is not enabled by default and requires prior human authorisation in order to be applied. Whether automated responses are actually used, even when available, is the focus of the next section, which considers the views of the security community towards automated response.

4.3 Attitudes towards automated response

In order to reflect the views of the security community towards automated response, Lee (2001) conducted a survey, as a complimentary project to the research described in this thesis. The survey was sent to IDS vendors and security specialists, requesting their views. In addition to the replies received, further research was carried on to enhance the findings from the survey. As a result of this process, it was found that the response capabilities of systems, in their current form, do not inspire a great level of confidence within the security community. Indeed the problem of false alarms in IDSs is substantial and a main concern, posing a significant obstacle in the adoption of automated response methods. In fact, as has been shown by Lee (2001), network administrators, and security specialists do not trust a system to issue automated active responses, even if they are available as a countermeasure. This finding is firmly supported and emphasised by recent online discussions that were initiated as a result of the introduction of automated response and prevention.

In the survey by Lee (2001), various IDS vendors and intrusion detection specialists expressed reservations regarding the issue of automated response. Representative of this was the view of Greg Shipley, security consultant and contributing editor for Network Computing online journal (Shipley 1999):

> *"Right now, in its current form, I don't believe that the current products are mature enough to be performing active response. ... any device that is re-configuring infrastructure equipment (shunning) could easily be turned into a denial of service tool."*

Shipley's views have not changed significantly since then, as in a more recent article, published in Network Computing journal (Shipley 2003), he said:

> *"If a marketing message's success can be judged based on its ability to confuse en masse, I think we'd have to award the "intrusion prevention" craze top honors. Although host-based intrusion-prevention systems hold promise, some network-based intrusion-prevention systems are disasters waiting to happen--repackaged intrusion-detection systems with published claims just short of an FTC violation. .... Truth be told, the message is both sexy and horribly misleading. These products* (network IPS) *don't eliminate your vulnerabilities, they just help stop certain types of attacks. Although there's nothing wrong with a tactical solution that adds a layer to your defenses, let's call a spade a spade: This isn't revolutionary technology; it's evolutionary, and its mutation is far from over."*

The dangers of active response in case of false alarms are also highlighted by security specialists Eugene Spafford from University of Purdue and Andy Cuff (Talisker), in their responses to Lee's survey (Lee 2001):

> *"Proactive measures are a reasonable idea, unless they can be subverted. For instance, if you decide to shut down your network connections as a proactive approach, then an intrusion attempt can be used as a denial of service"*
>
> Gene Spafford, <spaf@cerias.purdue.edu>

> *"As well as alerting to an attack occurring some IDS can defend against them, this is achieved in a variety of ways. Firstly by injecting packets to reset the connection or alternatively by reconfiguring routers and firewalls to reject future traffic from the same address. There are problems with both methods in order to inject packets the IDS needs to have an active interface thereby making itself susceptible to attack. See stealth (sic). There are ways around this, such as having the active interface inside the firewall. As for the latter method of automated response, it isn't unknown for attackers to abuse the latter method by spoofing the address of a friendly party and launching an attack, the IDS then configures the routers/firewalls to reject the these addresses, effectively DOSing themselves. "*
>
> Andy Cuff, http://www.networkintrusion.co.uk

According to Bill Oliphant, Product Manager of Intellitactics Inc., many of the security professionals in North America prefer not to use automated response systems, even if such methods are available. Thus their product, Network Security Manager (NSM), did not provide any automated active response mechanisms at the time.

> *"our product at this time does not provide a particular response mechanism out of the box,... What I have found is that, at least in North America, many of the security professional prefer not to use automated response systems. For the network management world, it is relatively easy to dictate when some action is required based on hardware or software failure. In the security world we must assume that the information that is provided, is merely untrusting information, and the degree of event correlation is paramount. The more you can correlate*

> *the events the easier it is to analyse the degree of threat. So it then is very difficult to run automated response, shut down applications, change router settings, prevent access to the Internet. I am sure that you have heard of the Hacker technique of creating "noise". It is very difficult to determine noise and the real thing. If shutting down or diverting services is the response then the hacker has won. The prime principles of confidentiality, integrity and Availability are destroyed from response type."*
>
> Bill Oliphant, Dir Product Management, Intellitactics Inc.,
>
> Network Security Manager (NSM)

In fact, even now, Intellitactics takes a more cautious view towards automated response, as according to their recent product literature (Intellitactics 2003), none of the active responses are initiated automatically; they need prior human authorisation in order to be applied. Specifically, NSM allows administrators to manually invoke third party tools, such as network scanners, sniffers, traceroute, and whois, to collect more information about an incident and if a threat is verified, the administrator can invoke vulnerability scanning tools against potential victims / threats / targets, or even block specific IP addresses on critical systems only.

SHADOW did not (and still does not) offer any automated incident response mechanisms either, due to the unreliability of automated response, according to Robert Blader, Information System Assurance Officer in the Naval Surface Warfare Centre.

> *"In short, there is no automated incident response mechanism built into SHADOW. Basically the issues are spoofed/anonymized source addresses and what action to take. I've read (a posting from H.D. Moore) where at one time one could use the TTL field to determine (most times) if an IP is spoofed or not. But shortly after, Nmap (and probably other tools) began randomising values*

> *put in TTL fields so that was no longer reliable. Plus, even if one can determine the real source, one is still left with the problem of what action to take. Retaliation (DoS for instance) is out of the question since one can't ignore the impact on innocent users coming from the same network (say with an ISP). Therefore, SHADOW will continue to only support detection and reporting."*
>
> Rob Blader, Information System Assurance Office, CD2S,
> Naval Surface Warfare Center

A partial approach to automated response had been adopted by Enterasys Networks. The view of the Sales Dept. at Enterasys Networks (formerly Network Security Wizards) was the following:

> *"Dragon does support automatic responses by allowing the end user to run programs via a tool we call 'AlarmTool'. These programs can be written by the end user or commercial programs (such as emailers, pagers, and other notification methods). Some customers do effect the configurations or rules of routers and firewalls.*
>
> *We at Enterasys (Dragon IDS) do not advocate the use of this tool to change rules on routers or firewalls. We believe that you can easily cause a DOS on yourself, and are especially bad if this is what the attacker wanted in the first place. We think it is dangerous to put all your faith in automatic responses believing that you are protected. Attackers are very smart and know how to use your own equipment against you if it will benefit their attack. We supply the end user with forensics of the alert (i.e. you can replay a session, look at the raw data packet, pull in firewall logs for correlation). We believe that the ability to see if an attack was successful or not and then have a human acting on that is better for the overall health and security of the network.*

> *We do have a tool that will shut down particular sessions, but will not block that IP from acting again. We consider this a compromise between automated response and no response at all. ...most people in the security product industry agree that automatic responses can be very dangerous and should not be relied upon to make important decisions about your networks."*
>
> Sales at Enterasys Networks (formerly Network Security Wizards)

Enterasys has changed direction since then, and has adopted a wider variety of active response actions, including IP blocking via firewalls or routers. However, that could be attributed to the recent drive in the industry to adopt intrusion prevention and response, rather than their change of views about the maturity of automated response. Indicative are the personal comments of Gary Golomb, Senior Research Engineer from Enterasys, made in a recent posting at an IDS mailing list (Golomb 2003).

> *"In private communications with Stiennon (the Gartner analyst)* (author of the 'IDS is dead' report), *he offered the shocking fact that - for all that they are hyping IPS - the team at Gartner 'doesn't know anyone who is using an IPS in inline mode.' That runs utterly contrary to the perception they are trying to create that IPS is the 'wave of the future' ... Anyways, the best solution for one environment is not going to be a market-wide best solution. I'm glad there are several other vendors who haven't completely succumbed to money-making hype* (of intrusion prevention) *and are taking a more responsible approach to researching these technologies. It's nice to see, and I applaud them all.*"
>
> Personal Comments of Gary Golomb, Senior Research Engineer, Enterasys

More detailed comments of IDS vendors and intrusion detection specialists about automated response, who replied to Lee, are included in Appendix A.

The reservations of the security community towards automated intrusion response shows that there is still great potential for improvement in the area. Response methods should be enhanced to take account of the fact that detection is not always perfect. Additionally, it must be possible to respond to a variety of incidents, rather than just a small subset of them. Therefore, it should be possible to offer scalable response options for a variety of certainty levels, and after considering a variety of other contextual factors, make more informed decisions that will enable the response decision engine to operate autonomously.

4.4 Current State of the Art in Response Technologies

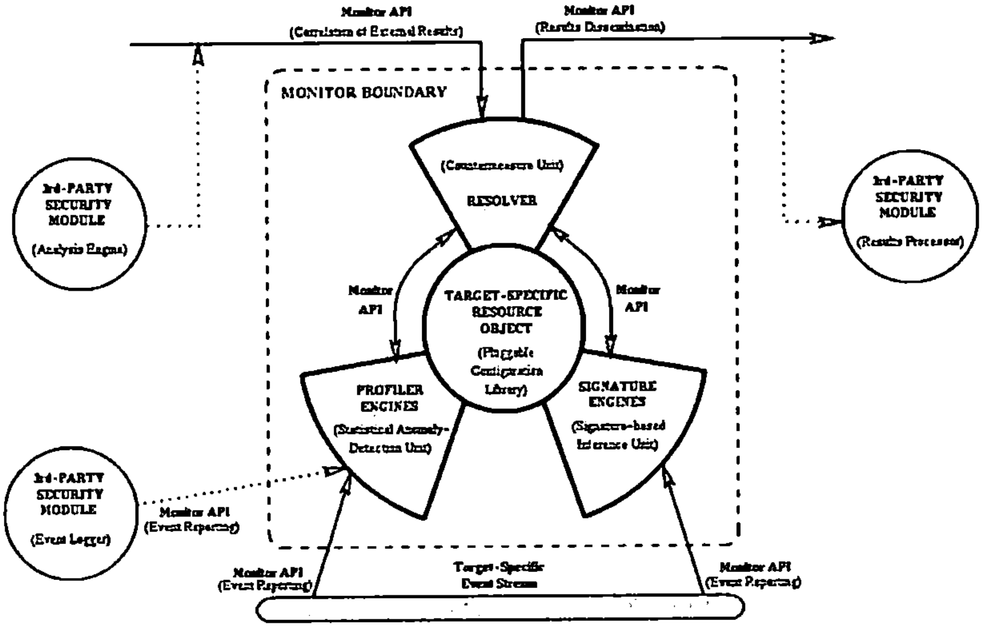
Challenges
1. Respond to a larger set of attacks
1a. Consider false alarm probability
1b. Support a wider variety of response actions
1c. Support response actions that can achieve more goals than stop / block the attack (e.g. investigate an incident, minimise damage at the target, forestall potential incidents, etc). In other words, support response actions that can be appropriate for cases of false alarm scenarios
1d. Support scalable levels of response

Table 4.1 Challenges for automated response

The previous three sections identified the main challenges of automating response in Intrusion Detection and Prevention Systems, showing that it is still a prime area for research. These challenges include the ability to respond to a wider variety of attacks, and not just the ones with low false alarm probability. This challenge relies on the ability of a response system to deal with false alarms, and support a wider variety of response actions, which can achieve more goals, other than permit or deny suspected events (for example, actions that aim to investigate incidents, minimise potential damage at the target, forestall potential incidents, etc). The challenges described are summarised in Table 4.1. With this in mind, this section describes the current research efforts aiming to address these challenges, and improve the response capabilities of IDS systems in general. A discussion about these research projects, and how they relate to the challenges for automated response is provided in section 4.4.7.

4.4.1 EMERALD Project

Event Monitoring Enabling Responses to Anomalous Live Disturbances (EMERALD) is an intrusion detection project being pursued within the Systems Design Laboratory at SRI International, which has been actively involved in the intrusion detection domain since 1983 (Porras and Neumann 1997).



Source: http://www.sdl.sri.com/

Figure 4.3 EMERALD Generic Monitor Architecture

EMERALD supports a decentralised architecture, with *domain-wide* and *enterprise-wide analysis*, covering misuse and anomalies across a single or multiple domains. The focal point of the architecture is the generic *Monitor* (Figure 4.3), which can be deployed as a *Service*, *Domain*, or *Enterprise-layer Monitor*. Service Monitor is a distributed component, providing localised, real-time analysis of the network infrastructure and network services. Domain monitors correlate reports from individual Service Monitors within the domain, and Enterprise-layer Monitors perform inter-domain event analysis, aiming to protect from information warfare-type attacks against the entire enterprise. The *Resolver*, as depicted in Figure 4.3, is the response module in the generic Monitor architecture, an instance of which is included in each Monitor. The Resolver is a countermeasure decision engine capable of fusing the alerts from its associated analysis engines and invoking response handlers to counter malicious activity. The Resolver is an expert system that receives the intrusion and suspicion reports from the *Profiler* and *Signature Engines*. Based on a combination of these reports with reports from other peer monitors, it decides what response to invoke, and how to invoke it. Possible responses may include direct countermeasures such as closing connections, terminating processes or the dispatching of integrity-checking handlers to verify the operating state of the analysis target. Another responsibility of the Resolver is to handle the interface with the security administrator (Porras and Neumann 1997).

EMERALD provides an interesting architectural approach, focusing upon the co-operation of distributed response elements. Porras and Neumann (1997) provide a very detailed discussion about the structure of the proposed system, describing a comprehensive approach to intrusion detection and response, and envisaging the use of active response methods for the counteraction of malicious activity. However, although the first implemented version of EMERALD is already available, research has so far focused upon the detection aspect of the project, rather than on the response issues. Hence few references have been made in the publications about the expert system of the Resolver, the Response Policies adopted, or the full set of response actions available.

4.4.2 Response and Detection Project

This research project is being pursued with the collaboration of Boeing Corporation, Trusted Information Systems, and UC Davis University (UC Davis 2000). It is an effort to combine IDSs with firewalls and routers to form an Intranet wide automated response system. The basic idea of this project is to enable the co-operation among response components in a virtual security network, where security components will alert each other of the attack and a component will be selected to initiate an automated response. Automated responses mainly examine network-based attacks and at the moment are restrained to just filtering network packets. However, focus has been given to the extension of response options, giving consideration to novel response actions, such as:

- introducing delay to network connections;
- a 'transfer to jail' system (an option which is not specifically explained, but is assumed to mean isolation of a machine or process that has been compromised);
- replacing sensitive files with look-alikes.

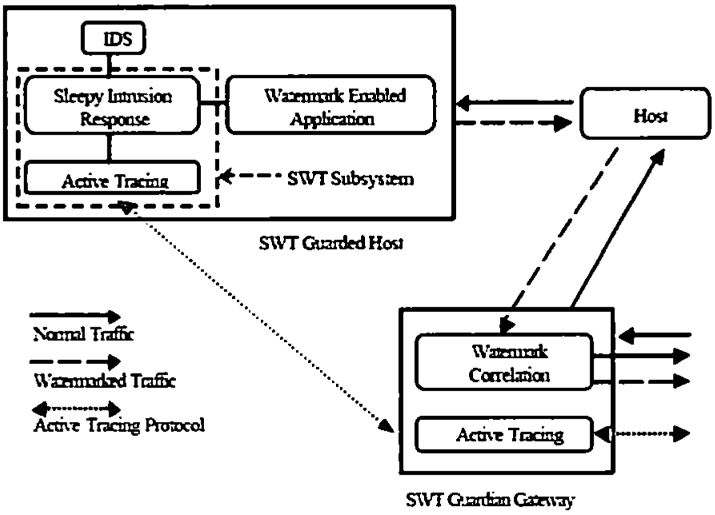
According to Dan Schnackenberg from Boeing Corporation (see Appendix A), the response options of the Response and Detection project to specific attacks are summarised below:

> *"We have not published anything on what an appropriate response would be to specific attacks. Our implementation has a few beliefs built into it, but those are not documented outside the code. Those beliefs are quite simple:*
>
> *(1) probes deserve to be traced and reported, but no blocking rules are applied at firewalls or filtering routers;*
>
> *(2) floods deserve to be traced, reported, and stopped using rate limiting mechanisms in filtering routers/firewalls; and*
>
> *(3) other attacks deserve to be traced, reported, and stopped using packet filtering rules.*
>
> *We also developed a capability to attempt isolation of a machine that has been compromised. Finally, we developed some host-based mechanisms that attempt to respond to host-based detector alerts by performing actions such as killing a user's session or disabling a compromised user account."*

For this project, Boeing Corporation has supplied the intelligent routers, as well as the Intruder Detection And Isolation Protocol (IDIP), which enables the components to communicate with each other. The Master Intrusion Detection System (MIDS) was used in this project and was supplied by UC Davis. Its distributed nature and the adoption of innovative response actions, that include tracing of attacks, are the most important aspects of this project, whereas little focus has been given on the response decision mechanism itself.

4.4.3 Tracing Based Active Intrusion Response Project

Another approach that includes tracing is the Tracing-based Active Intrusion Response (Wang et. al. 2001). In this case, the response mechanism is focused around the ability to identify the source of intrusions, which need to be bi-directional and interactive in order to be traced. The tracing technique, *Sleepy Watermark Tracing (SWT)* (Wang et. al. 2001a), is initiated whenever an intrusion is detected and involves the injection of a watermark into the backward traffic of a suspicious connection, aiming to trace the connection back to its source. Each watermark is randomly generated and can uniquely identify each connection. As a result, it has been designed to be large enough to avoid other randomly generated watermarks being identical (if the watermark is 73 bits then the probability of a collision is less than 0.1%).



Source: http://www4.ncsu.edu:8030/~xwang5/

Figure 4.4 Sleepy Watermark Tracing (SWT) Architecture

The main components of the Sleepy Watermark Tracing Architecture are depicted in Figure 4.4, and include the *Sleepy Intrusion Response (SIR), Watermark Correlation (WMC)* and *Active Tracing (AT)*. After the detection of an incident, the SIR initiates and co-ordinates the active tracing, with the AT and *Watermark-Enabled Application*, assigning a unique watermark for the connection and injecting it to the backward traffic respectively. Each SWT-enabled gateway, that receives the message, will firstly determine information about the next leap in the connection chain (by correlating the incoming vs. outgoing connections with the same watermark) and then send trace information back to the original host that initiated the trace. It will also notify the next gateway to initiate the tracing procedure as well.

After tracing the intrusion, responses can be applied nearer to the source of the intrusion, making them more effective. Some responses that could be applied, include:

- **Remote monitoring and surveillance.** By enabling the closest to the source SWT-enabled gateway to monitor the intruder, it is possible to report his activities back to the target, and maybe reveal information about other compromised hosts.

- **Remote decoy and trap.** The purpose of this response is to give the intruder the impression that he is still connected to the target host, whereas he has been diverted to a decoy system. In the meantime, the decoy can intentionally introduce delays to keep the intruder occupied for as long as possible.

- **Remote blocking and containment.** Blocking occurs at several points towards the intruder, aiming to completely contain him and prevent other targets being attacked.

- **Remote isolation and quarantine.** Compromised hosts are isolated, until they are recovered.

The implementation of SW Architecture requires specially designed gateways, using active networks technology, and thus it could not be easily applied without modifications to the existing network infrastructure. Also, it can be suitable for tracing only a subset of intrusions, such as unauthorised access attempts; Denial of Service attacks could not be traced, as they do not require any feedback from the target, and hence, do not use bi-directional interactive connections. Having said that, SWT only relies on network elements, such as routers and switches, and thus, it overcomes the problem of relying on potentially compromised hosts to perform tracing and issue response actions. Given that the probability of a router being compromised is very small (as according to CERT/CC (Howard 1997), it is computer hosts, rather than network elements, that get subjected to unauthorised-access incidents), SWT has the advantage of being more accurate and reliable. Finally, the use of active networks technology enhances the response functionality, as some responses could not be easily implemented in different environments (for example, redirecting the connection to a decoy and trapping the intruder).

4.4.4 Automated Response Broker (ARB)

This research effort (Balepin et al. 2003), from the Security Laboratory of University of California at Davis and the Bauman Moscow State Technical University, aims to address problems associated with automated response in host-based, signature-based intrusion detection systems, and specifically the SHIM intrusion detection system (Ko 1996). It tries to overcome a significant problem of commercial response and prevention systems, which is the lack of escalating response, by applying varying levels of response actions, according to the risk introduced by the intrusion. The aim is to avoid issuing unnecessary and costly response actions, which end up causing more damage than the intrusion itself, especially in the scenario of false alarms. It considers a cost-based model, which associates all system resources, and response actions to specific costs, trying to form response decisions based on the balance between the cost of the resources threatened by the intrusion, and the gain associated with the possible responses. As a result, some intrusions might be stopped or restored partially, up to the point where the system considers it is worth acting.

Upon the generation of an intrusion alarm, the automated response system initially freezes all suspect processes at the host, and their children, to prevent the damage from the intrusion escalating even further. The system then tries to restore the system to its initial state, by considering all the suitable response actions for the given circumstances, trying to balance the cost associated with each possible response action against the cost of doing nothing. After considering all the options, the responses that combine minimum cost and maximum gain are preferred over the others, which have the potential to endanger the system even further.

The main advantage of ARB is the fact that it addresses a significant problem of automated response; it offers a variety of response levels that are not limited to the "either permit or deny" approach of commercial IDS/IPS systems, and it tries to select varying levels of response actions, according to the costs associated with them, ensuring that a response action will not introduce more damage than the intrusion itself.

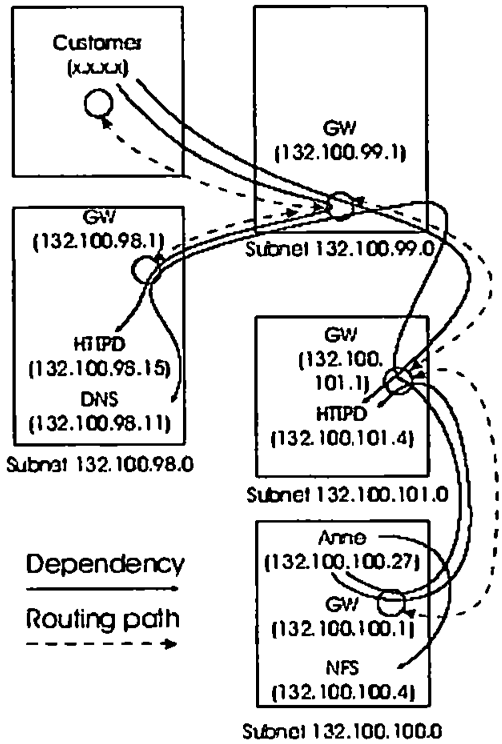
A limitation of this approach is the fact that it considers only signature-based alerts in a single host and thus it does not cover state-based or anomaly based alerts, nor network based attacks. Also, it considers only response actions that aim to either stop the attack or restore the system to its initial state, whereas automated responses could be usefully employed to collect more information about the incident, or identify the attacker. Also, a wide variety of response actions, which include deception and tracing of attacks, could also be usefully employed to protect system resources and stop attacks. Finally, the decision capability of a responder could be further enhanced by considering a wider variety of influencing factors to determine the threats and costs associated with attacks, responses and targets; some examples include the load on the system at the time, or whether the frequency of the occurring attack (the dynamics) is unusually high. Also, the system does not take into account the probability of a false alarm, which could end up disrupting legitimate users.

4.4.5 Automated Intrusion Response Model

This research effort, from the Technical University of Vienna (Toth and Kruegel 2002), is another cost-based approach, which focuses upon evaluating the impact of automated response mechanisms on network resources and their users, aiming to determine which of the appropriate responses has minimum impact and thus can be preferred over the others. The response actions supported by this model include firewall and process based responses, such as updating of firewall rules, killing and restarting of processes and disabling / enabling of user profiles at hosts.

The assessment of the impact associated with each potential response is done via a complex model, which takes into account the network topology (in the form of routing tables and firewall rules), the dependency relationships between different resources, and the importance of these resources to different users in the network. The cost of each possible response is calculated by considering the importance of the resources it can affect by making them unavailable. The model will track any changes in the network, which might occur as a result of response actions, in order to update the dependencies between resources and reflect the status of the network dynamically. For example, changes due to the reconfiguration of a firewall, or the (un)availability of services to specific users will be taken into account dynamically.

The role and functionality of the model can be illustrated by considering a network of 4 subnets (see Figure 4.5), where the HTTP server (132.100.101.4) is subjected to a DoS attack from the Internet. The response system might decide to block the outside traffic either at the external gateway (132.100.99.1) or at the gateway on the same subnet as the attacked server, depending upon which option has lower cost. The dependencies for one of the users (Anne) are depicted in Figure 4.5, where it is shown that she depends upon the attacked HTTP server, the Network File Server (NFS), the DNS server at subnet 132.100.98.0 and all the gateways connecting them. The Internet customers depend upon the attacked HTTP server and on the HTTP server at subnet 132.100.98.0. Blocking the external traffic at the external gateway (132.100.99.1) will affect the access of external users to the second HTTP and DNS server, whereas blocking it at the 132.100.101.1 gateway will not have that effect. In both cases, external traffic is blocked on the attacked server, preserving its secure state and the related users (in this case Anne) are not affected at all by any of the response options. By considering the impact of each alternative response scenario on the affected entities, the responder will finally select the one with the lowest negative effect, which in this case is the blocking of external traffic on the internal gateway 132.100.101.1.



Source: http://www.infosys.tuwien.ac.at/Staff/tt/publications/

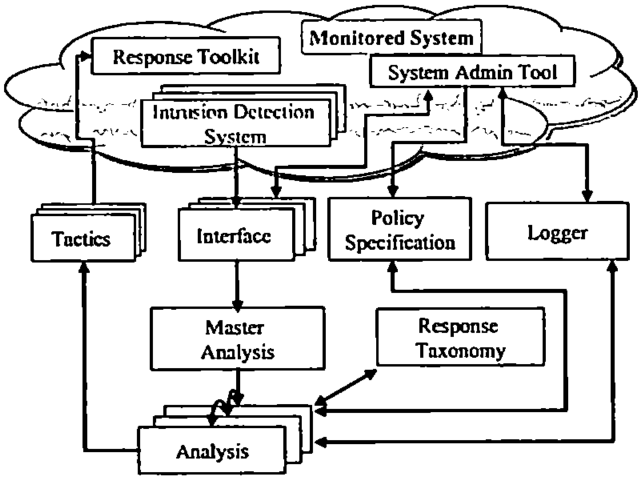
Figure 4.5 Topology and Entity dependencies

The main contribution of this approach is that it can reflect changes in the environment, determine the impact of responses dynamically, and try to select the ones with the less significant effect. This approach embodies the essential aim of a response system, especially in cases of false alarms; that of issuing responses, which will preserve the security status of the system at the least possible cost. However, considering only the dependencies between the affected resources does not always enable the consideration of other influencing factors. For example, this model does not take into account factors such as the existence of highly vulnerable software (e.g. SQL Server) or auditing software on the target, which although it cannot be represented in the dependency list (as none of the affected entities depend upon it), can increase the threat of the attack significantly, by facilitating its progress even further. Also, it does not take into account the speed of the attack and the timeline in which a response should be needed. For example, in case of a virus, which is likely to escalate very quickly, responses should be immediate and with higher impact, if necessary. However, in cases of slower attacks, responses could be initiated to run in the background, without affecting any users. Alternatively, severe responses would not be issued, unless authorised by an administrator.

4.4.6 Adaptive, Agent-based, Intrusion Response System (AAIRS)

The most comprehensive approach to an intrusion response methodology is presented by the AAIRS project, which is being pursued within the Computer Science Department of Texas A&M University (Carver et. al. 2001). It focuses upon the response decision mechanism, proposing a methodology for adaptive automated response using intelligent agents (see Figure 4.6).

In AAIRS, a new *Analysis* agent is created every time a new attack is reported by the IDS to the *Master Analysis* agent. The *Analysis* agent develops an abstract response plan for that attack, based upon the *Response Taxonomy* agent and the *Policy Specification* agent that will determine a response goal and limit the response based upon legal, ethical, institutional, or resource constraints. The *Analysis* agent then passes the selected course of action to the *Tactics* agent. The *Tactics* agent decomposes the abstract response plan into very specific actions and then invokes the appropriate components of the *Response Toolkit*. Both the Analysis and Tactics agents employ adaptive decision-making based upon the success of previous responses. Finally, the *Logger* records Analysis and Tactics agents' decisions for system administrator review.



Source: (Carver et. al. 2001)
Figure 4.6 AAIRS System Architecture

Emphasis has been given to the limitation of uncertainty in the response decision process (Carver et. al. 2001), as well as the adaptation of the system (Ragsdale et. al. 2001) based upon the effectiveness of its detection and response capability in the past. For the response decision process, the following factors have additionally been taken into account (Carver and Pooch 2000b):

- **Timing of the attack**: Different responses will be selected, based on whether they have to be issued prior, during, or after an attack. Responses prior to an attack will be pre-emptive, aiming to increase the defence of the potential target. Responses issued during an attack will aim to control the damage caused, by limiting the effect of the attacker on the system. Finally, responses issued after an attack aim to document and repair any damage to the system.

- **Type of attack**: According to whether the attack is a threat to the Confidentiality, Integrity, or Availability of targeted systems, different responses should be issued. For example, Denial of Service attacks require different actions than unauthorised access attacks (e.g. race condition attacks, which are associated with synchronization errors that provide a window of opportunity, during which one process can interfere with another, possibly introducing a security vulnerability).

- **Type of attacker**: Responding to a script-kiddie, who is using a well-known attack script, is different to responding to a distributed, coordinated computer attack supported by a military organisation. Hence, attackers are classified as Cyber-gangs, Economic Rivals, Military Organisations, etc.

- **Degree of suspicion**: Given the problem of false alarms, and the fact that some events are more prone to false alarms than others, the strength of suspicion should be taken into account to select appropriate responses. The levels of suspicion are represented as low, medium or high.

- **Attack implications**: Based on the importance of different systems within an organisation, attack implications can vary. For example, response should be different if a single workstation is subjected to a Denial of Service attack, rather than a central Domain Name Server.

- **Environmental Constraints**: This factor represents the legal, ethical, institutional, and other constraints that limit which responses are appropriate. For example, according to U.S. law, launching a counterattack against a suspected attacker is prohibited, unless the attack is part of a military operation that occurs during a declared war. Some of the environmental constraints are: No offensive responses, No router Resets, etc.

This approach is the most comprehensive so far, providing a useful methodology of looking at responses, and giving considerable focus on the mechanism and the influencing factors of the response decision process. However, the set of influencing factors could be extended even further to reflect aspects, such as the current status of the target at the time of the incident, etc.

4.4.7 Discussion

After considering the problems of automated response, the main contributions of research in the area are summarised as follows:

- The implementation of distributed response systems, such as EMERALD and the Response and Detection Project, able to protect large and complicated organisational networks. Although this characteristic does not directly relate to the challenges already identified, it is still relevant to consider it, as it provides an important basis for enabling automated response in real networks, and thus it indirectly contributes to the advancement of automated response.

- The focus on novel response actions, such as tracing the source of attacks. Although tracing network connections has been recognised as an extremely difficult problem, which often requires collaboration among different networks (especially if the connection originates from outside the organisation), it is still important in cases of internal attacks, or when evidence about the attacker needs to be collected. After all, its role could become more significant, if organisations worldwide agreed to allow tracing by IDS components in their systems. The challenge of supporting a wider variety of response actions has not been fully explored with the existing efforts, as not much focus has been given to actions that aim to investigate, or forestall future incidents.

As for enhancing the response decision mechanism specifically, the main contributions in the area are summarised below:

- The ability of a system, such as the Automated Response Broker, to issue scalable responses, according to the risk introduced by the intrusion, and the effect of the response. That overcomes the problem of existing IDS/IPS solutions, which can either permit or deny a security event, and do not offer any flexibility required for cases of false positive alarms. Moving one step further from ARB, it would be desirable to enable a responder system to support a wider variety of response actions (other than stop the attack and restore the system) that will offer varying levels of protection.

- The use of a cost-based model to determine the appropriate level of impact a response action should have by weighing the impact of the intrusion against the impact of the response. The aim of this approach is to minimise the counter-effects of response actions and avoid scenarios when responses cause more harm than the intrusions themselves. Automated Response Broker and Automated Intrusion Response Model have adopted a cost-based approach, and although it is a clear enhancement over current IDS/IPS approaches, it could still be extended even further, to consider a wider set of relevant factors that will increase the awareness of the responder system and enable it to select more appropriate responses.

- The ability of a system to deal with false alarms, as featured in AAIRS. This feature is very important for intelligent automated response, and can be enhanced even further by broadening the input variables of the response decision engine, which will enable the responder to make more informed and thus more intelligent decisions. The broadening of the input variables can be achieved by representing the context of an attack.

- The dynamic assessment of the response impact, in order to account for changes in affected systems and services at the time of the incident. Although the changes accounted for only results of response actions, such as disabling or limiting access to specific services, it is still an important feature that increases the awareness of the responder system. This approach, though, could be extended to reflect the current status of the target at the time of the incident, account for other changes in the target, such as the services / applications running, or its load at the time.

Finally, it is important to recognise the contribution of AAIRS, which uses a wide variety of contextual factors for the intrusion response mechanism, including the probability of a false alarm, the success of responses based on their history, and the importance of the target. AAIRS is the most complete approach so far, in terms of its response mechanism. The research presented in this thesis has similar focus as AAIRS, which is the effort to increase the awareness of an automated responder, and enable it to make as informed decisions as possible, eventually allowing it to operate autonomously.

4.5 Conclusions

This chapter has identified the need for automated response, the problems preventing its adoption, and the fact that the security community and administrators do not trust it. Existing research has addressed some aspects to improve the situation, but as has been discussed in section 4.4.7, scope for further research remains, if the system is to operate autonomously. The views regarding automated response, as expressed by the security community, and the limitations of existing research efforts have been used to inform the design of the new response architecture, presented later in this thesis. However, the first step towards furthering research is to systematically study intrusions and their characteristics. The next chapter presents the results of that study.

CHAPTER 5

A RESPONSE-ORIENTED TAXONOMY OF INTRUSIONS

5 A RESPONSE-ORIENTED TAXONOMY OF INTRUSIONS

5.1 Introduction

The need for an *Intrusion Taxonomy* can be illustrated by looking at the definition of the term itself. According to Amoroso (1999), the term *Intrusion Taxonomy* in an IT context is defined as:

> *"a structured representation of intrusion types that provides insight into their perspective relationships and differences."*

Focusing on the perspective relationships and differences of intrusions is an important part of understanding them and an especially significant step towards deciding how to react to them. Thus the use of a taxonomy of intrusive activities, which will focus upon the intrusion characteristics important for intrusion response, will enable the understanding of intrusions and move forwards the design of a response decision mechanism.

Since current intrusion classification taxonomies provide little understanding beyond the level of the incident types, or the general security impacts of intrusions, a new taxonomy has been developed as part of this research, aiming to consider incidents and identify their different results in different contexts. This taxonomy is intended to provide insight into the process of selecting appropriate responses, and forming the basis of decision-making in an automated responder system. Thus, its main objective is to demonstrate that the same incident can demand different responses in different contexts. This chapter begins by justifying the need for a new taxonomy, before proceeding to present details of the new approach.

5.2 The need for a new approach

Previous research has given rise to a number of intrusion taxonomies, each of which presents an alternative view of the situation. Brief summaries of a number of notable approaches are given in Appendix B.

Most of the existing taxonomies contribute to the systematic study of intrusions and can be considered suitably comprehensive and accurate, as they include an extensive number of classes. However, they all give insight into the issue from a general security perspective, without taking into account the areas of intrusion detection and response. From a detection perspective, it is clear that a number of the incident classifications (e.g. social engineering, physical tampering), and issues such as the objectives of attackers, could not be detected by an IDS. In addition, there is no specific focus upon the issue of response. A taxonomy that would focus upon intrusion detection and response ought to give consideration to the main response influencing factors, such as incident type, target, and/or potential impact. This will demonstrate how the same incident can have different results in different contexts, and thus require different responses. The outcome of this process will lead to the indication of generic response categories, considering what can be done to halt an attack in progress, reduce its impact and/or prevent reoccurrence. The discussion of such a taxonomy is the focus of the next section.

5.3 A Response-Oriented Taxonomy

In order to derive an appropriate taxonomy it is necessary to give consideration to characteristics of intrusions that can influence the response process (Papadaki et. al. 2002). The most evident characteristic that falls into that category is the result(s) of an intrusion, which is defined as the consequence(s) of a successful attack in a system. However, unlike previous result-based taxonomies (Cohen 1995; Russell and Gangemi 1991), the result is not represented in only one dimension, as there are multiple aspects encompassing the likely result(s) of an intrusion. Instead, the likely result(s) are comprised of *Urgency*, *Severity*, *Impact(s)*, and *Potential Incidents*.

The *Urgency* relates to the need for timely response, and partially reflects the speed of the attack. Since some attacks can evolve more rapidly than others, it is important to consider how much time is available to respond in each case. A Denial of Service attack, launched with the use of automated scripts is an example of a rapidly evolving attack, while probing systems to identify their vulnerabilities, allows a greater window of opportunity for response, as the incident is likely to evolve over a longer period of time (Honeynet project 2000b).

Another attribute of the result is the *Severity* of the intrusion, which relates to the magnitude or extent of the attack. The more severe an intrusion is, the more important it is to be contained, and eventually be eliminated, if the system is to recover from the incident. In the taxonomy, both *Urgency* and *Severity* are rated on a scale of Low, Medium, High for each incident / target combination. The three-point scale was chosen to show the relative differences in the results of intrusions, among various combinations of incidents and targets. These combinations are presented with a high degree of abstraction, and thus the ratings used had to be general as well. For example, the severity and urgency of scanning attacks can vary according to the specific type of tool used, however, most scanning attacks generally have, more or less, low to medium severity and urgency.

Another aspect of the result is the *Impact(s)* of an intrusion upon a system. The *Impacts* relate to the assets of the system that have been compromised by the intrusion, and may be observed and measured in relation to the Confidentiality, Integrity and / or the Availability of systems and data. Although in scenarios, such as conventional risk analysis (Davey 1991), it is normal to rate these impacts on a sliding scale to indicate their severity, the taxonomy in the table that follows simply indicates whether there is a potential impact or not, as assignment of values would be too subjective. Apart from the fact that the incident categories are too general for meaningful distinction of severity values to be made, an indication of the impact severity is provided via the *Severity* attribute anyway.

The final element of the result relates to whether any further incidents are likely to be facilitated as a consequence of the initial attack. This is expressed in the taxonomy as *Potential Incidents*. For example, when sniffing software is used to capture network traffic, it is likely that the information obtained (e.g. user names and passwords) will enable attackers to log in as legitimate users at a later date and thus succeed to masquerade. Also, in the case of a virus, the potential incidents could be denial of service, leakage of confidential files, or even damage to system and user files. In other words, the potential incidents indicate the threat that has been introduced within the system after the occurrence of the original incident. Such information is important, if the system is to pre-empt further attacks and avoid the damages escalating even further. Admittedly it might not always be possible to usefully forestall all potential incidents, as a specific attack could virtually lead to all types of potential incidents. In such case, the cost of preventing all of those would overwhelm the available resources. However, that should not limit the value of being able to pre-empt threats, whenever possible.

Since the same incident can have different impacts upon different targets, another important characteristic that can influence response is the *Target* of the intrusion, which is defined as the receiving end of the incident. The target groups considered in the Response-Oriented taxonomy are as follows:

- **External server**: Public-facing servers that are accessible from external networks and represent the public image of the host organization (e.g. web, email, DNS, FTP servers). Ideally, if configured correctly, external servers should not contain or facilitate access to confidential information, but ought to provide accurate and uninterrupted service to clients.

- **Internal server**: Servers accessible only within the internal network of the organization (e.g. intranet web and file servers). Information contained in internal servers has the potential to be confidential, and thus, apart from requiring accurate and uninterrupted services, they also need to preserve the confidentiality of their data.

- **User workstation**: Computing units used by average users, which are likely to contain information specific to a particular user and their role within the organisation. Apart from the potential to contain confidential information, user workstations can be targeted with the intention to be used in great numbers to carry out distributed attacks against other targets (NIPC 2001). Finally, their advanced processing, storage, or networking capabilities could prove sometimes to be another desirable asset for prospective attackers (Borland 2003).

- **Network Component**: Networking equipment such as routers, switches, firewalls, which may be targeted as a means of affecting other systems or subverting operations.

This is by no means a detailed or exhaustive list, but it is sufficient to give a high level of abstraction for the different elements that might be targeted in a typical organisation. Also, it should be noted that only the *type* of target has been considered in the taxonomy. The number of targets attacked, which reflects the scale of an incident, will also influence its severity, but it has not been considered. For example, a virus incident that infects a few user workstations is not as severe as one that infects the vast majority of them. However, considering the scale of the intrusion as well would add more complexity to the taxonomy, and would make it difficult to illustrate its main point. The primary aim of the taxonomy, at this stage, is to demonstrate the effect of different targets upon the results of intrusions, and thus it is considered that the level of detail already presented is sufficient to illustrate that main point.

In order to demonstrate the main concept of the Response-Oriented Taxonomy, a set of incidents has been used and is listed below:

- Information gathering (Probe / Scan, Sniff)
- Authentication failure (Masquerade / Spoof, Bypass)
- Software compromise (Buffer Overflow, Flood / Denial of Service (DoS))

- Malware (Trojan Horse, Virus / Worm)

- Misuse (Unauthorised Alteration, Unauthorised Access)

As with the previous taxonomies, the selection of incidents is by no means exhaustive, but the five top-level categories aim to represent the most significant set of incidents, included in knowledge bases of detectable intrusions, as released by IDS product vendors and incident response teams (Internet Security Systems 2001; CERT/CC 2003b, CERT/CC 2004). At the same time, it is important to preserve a high level of abstraction, with each incident type including as many cases of incidents as possible. So, for example, although there are many different methods of launching Denial of Service attacks (such as SYN Flooding, SMURF attacks, Ping of Death, Trin00, and others), their ultimate effect upon a system is similar, and it is this that will be the main determinant of the desired response(s). Still, it should be noted that had each of these attacks been rated individually, their ratings would vary slightly, even if they belonged in the same category. However, the description of generic incident categories was considered appropriate to serve another purpose of the taxonomy, which is to lead to the indication of generic appropriate responses.

Having introduced the top-level elements of the taxonomy, the focus now moves to the five incident categories, as well as justifications to accompany the various ratings included in Table 5.1.

INCIDENT	TARGET	RESULT					
		URGENCY	SEVERITY	IMPACT			POTENTIAL INCIDENTS
				C	I	A	
1. Information gathering							
Probe / Scan	External server	Low	Low	✓		✓	Spoof, Bypass, S/w compromise, Malware
	Internal server	Medium	High	✓			
	User workstation	Medium	Medium	✓			
	Network component	Low	Low	✓		✓	
Sniff	External server	Low	Low	✓			Masquerade, Bypass, S/w compromise
	Internal server	High	High	✓			
	User workstation	Medium	Medium	✓			
	Network component	Medium	Medium	✓			
2. Authentication failure							
Masquerade / Spoof	External server	High	High	✓		✓	Misuse, Malware, Software compromise
	Internal server	High	High	✓			
	User workstation	Medium	Medium	✓			
	Network component	High	High	✓		✓	
Bypass	External server	High	Medium	✓			Misuse, Malware
	Internal server	High	High	✓			
	User workstation	High	Medium	✓			
	Network component	High	Medium	✓			

INCIDENT	TARGET	RESULT					
		URGENCY	SEVERITY	IMPACT			POTENTIAL INCIDENTS
				C	I	A	
3. Software Compromise							
Buffer Overflow	External server	High	High		✓	✓	Bypass, DoS, Misuse, Malware
	Internal server	High	High		✓	✓	
	User workstation	High	Medium		✓	✓	
	Network component	High	Medium		✓	✓	
Flood / DoS	External server	High	High			✓	Spoof
	Internal server	High	High			✓	
	User workstation	Medium	Medium			✓	
	Network component	High	High			✓	
4. Malware							
Trojan Horse	External server	High	High	✓	✓	✓	Bypass, Misuse, Malware, S/w compr., Info. gathering
	Internal server	High	High	✓	✓	✓	
	User workstation	High	High	✓	✓	✓	
	Network component	High	High	✓	✓	✓	
Virus / Worm	External server	High	High	✓	✓	✓	Misuse, Malware, S/w compr., Info. gathering
	Internal server	High	High	✓	✓	✓	
	User workstation	High	High	✓	✓	✓	
	Network component	High	High	✓	✓	✓	

INCIDENT	TARGET	RESULT					
		URGENCY	SEVERITY	IMPACT			POTENTIAL INCIDENTS
				C	I	A	
5. Misuse							
Unauthorised Alteration	External server	High	High		✓	✓	Malware
	Internal server	High	High		✓	✓	
	User workstation	High	Medium		✓	✓	
	Network component	High	High		✓	✓	
Unauthorised Access	External server	High	Low	✓			Malware, Unauthorised Alteration
	Internal server	High	High	✓			
	User workstation	High	Medium	✓			
	Network component	High	Low	✓			

Table 5.1 Response-oriented Intrusion Taxonomy

5.3.1 Information Gathering

The main characteristic of these intrusions is that they collect information about a target, aiming to identify exploitable vulnerabilities. Although information gathering does not have significant impact upon a system, it carries the danger of the knowledge gained subsequently being used for launching other attacks with higher severity. Probe, Scan and Sniff are intrusions that fall into that category and are described below.

5.3.1.1 Probe / Scan

Probe is used to access a target in order to determine its characteristics. Scan, on the contrary is used to access a set of targets in order to determine which of them have a specific characteristic (Northcutt 1999). The characteristics in question relate to the architecture of targeted systems and networks, such as their network configuration, or specific versions of services and operating systems. Knowing the version of a specific service, for example, enables the attacker to exploit the vulnerabilities associated with that software and thus achieve access to the system more easily (i.e. rather than blindly trying to exploit vulnerabilities, which might not even exist in the targeted system). *Potential incidents* after the occurrence of probing / scanning include more or less any kind of vulnerability exploits that gain access to a system, such as spoofing, bypassing authentication, compromising software and introducing malware. The *impacts* relate to breach of confidentiality, as information is obtained without authorisation. Probing and especially scanning can also degrade availability in some cases, by producing large amounts of traffic when probing / scanning multiple targets. External servers, as well as network components, can be affected in this manner, as in both cases availability is highly important and it is those targets that are more likely to deal with that traffic anyway.

An example that can illustrate the impacts of scanning is the Code Red II worm (Internet Security Systems 2001b), a variation of the Code Red worm, which has the ability to scan for vulnerable web servers very quickly, and consequently overload network components by producing large amounts of traffic. The *severity* of scans / probes varies, depending upon which target it is directed to. In the case of external servers and network components, which are genuinely subjected to unknown and thus untrustworthy users, they should be designed to be more tolerant to attacks of this nature. After all, they often provide the same nature of information within their normal activity anyway. Thus the severity of probing / scanning is not significant in those two cases. The *urgency* to respond is equally low, as probing / scanning is not likely to escalate rapidly (Honeynet Project 2000b).

On the contrary, probing or scanning an internal server is not usual, as it might be an indication of the existence of an already compromised system within the network. Thus it raises higher level of suspicion, which makes the level of high *severity* more appropriate. The *urgency* to respond is medium, due to the high level of suspicion on one hand, and its slow nature, in terms of escalation, on the other. As for user workstations, although probing / scanning a user workstation is also rare and thus should raise high level of suspicion, its impact is not as severe, as the threat to confidentiality in this case is significantly lower. Confidential information, if there is any, is not really threatened by inquiring about the versions of software running on the system. Thus the severity can be regarded as ‘medium’. However, the occurrence of such an incident could mean prior breach of another target (e.g. DNS server), and thus a medium level of urgency to respond is considered appropriate.

5.3.1.2 Sniff

Sniffing consists of the interception of traffic while it travels across the network. It is achieved with the use of software tools that can capture network packets either locally (within the same LAN) or remotely. The latter can be achieved by obtaining unauthorised access to monitoring tools that may be used to monitor traffic on remote network segments for legitimate purposes, but fail to provide adequate internal access control mechanisms (Internet Security Systems 2001). The range of information obtained with sniffing can often be more valuable than in the case of probing / scanning, as it could be anything that travels across the network, such as user name and password combinations, data files, and system or network information. In case of unencrypted information, the extraction of information is much easier (e.g. telnet packages). However, even encrypted data could be extracted after using readily available cryptanalysis software. After obtaining information with sniffers, the potential incidents that are likely to follow can mainly be masquerading, bypassing, and compromising software.

The impacts of sniffing mainly involve loss of confidentiality, but its severity and urgency depend upon the type of targets to which it is subjected. In external servers the *severity* is low, since again the nature of information disclosed should not be significant enough to raise the level of severity. Similarly with probing / scanning, the need for timely response is low, since the chance of rapid escalation is low.

In the case of internal servers, the *severity* is again high, but the need to respond is also high, since the nature of information that can be disclosed in this case is more significant and thus requires a more urgent issue of response. As for user workstations, the nature of information exposed might be significant, although not as significant as in the case of internal servers. So, both *severity* and *urgency* are considered to be medium.

Finally, in the case of network components, the *severity* of sniffing is medium, since the nature of information exposed in this case (e.g. Access Control Lists, administrator user account details) is significant enough to raise the level of severity (At this point it should be noted that, although not often, a network component could transmit this sort of information as part of a remote configuration session, via telnet, or HTTP). The *urgency* to respond is medium as well, since network components often represent single points of failure, and a possible compromise would affect multiple hosts.

5.3.2 Authentication Failure

Users and processes need to identify and authenticate themselves quite often in order to be granted specific access privileges. The most traditional method of authentication involves the use of user names and passwords that are usually required at the beginning of every session. The simplicity of this approach has made it very popular, even if it is not the most effective authentication method and does not provide the highest possible level of protection (Furnell et. al. 2000b). As a result, defeating the authentication process is very common, and can be summarised in three main ways, namely Masquerading, Spoofing and Bypassing.

5.3.2.1 Masquerade / Spoof

Masquerade is the action in which valid identification and verification information that belongs to legitimate users is obtained and used by an impostor. For example, an attacker might use a sniffer to capture user name, password and IP address combinations that are sent across the network, and then use this information to log into accounts that belong to other users. Spoofing, by contrast, involves the provision of false information. In network communications, each packet of information travelling on a network contains source and destination addresses either in the form of MAC, IP addresses, TCP connection IDs, or port numbers. Supplying accurate information is often assumed, but it is possible that incorrect information is entered into these communications, in order to accept an impostor address as original, and either have the impostor trick other machines into sending data to an innocent target or enable him to receive and alter data. Examples include IP spoofing, email spoofing and DNS spoofing. IP Spoofing attacks involve the use of IP packets with false IP Source addresses, which aim to either hide the true identity of the attacking source in order to gain unauthorised information (or access) at the receiving host, or subject the spoofed IP address in a third-party attack (e.g. Denial of Service attack) (CERT/CC 2000). Email Spoofing attacks involve the use of false email addresses as Sender, aiming to hide the true identity of the sender and usually obtain unauthorised information or trick the receiver of the email into facilitating the process of another attack (CERT/CC 2002b). Finally, DNS Spoofing involves the "impersonation" of a DNS Server, which leads true DNS servers to cache false records, and consequently result in legitimate users visiting false sites (CERT/CC 2002c).

Masquerading and spoofing are mainly a threat to the confidentiality of systems, since they most often provide unauthorised increased access to attackers. However, in the case of external servers and network components, it is possible to cause loss of availability as well, if used as a technique to enable the occurrence of DoS attacks. An example of such case is using IP spoofing to create half-open TCP connections (SYN Flooding), where the spoofed source IP is a host unable to reply. The target (usually web, ftp, email servers, even network components with TCP services enabled) will eventually fill its table of pending connections, making it unable to accept any others. If the attacking system keeps initiating half-open connections at a rate greater than the target can expire the older pending ones, SYN flooding can result in making the target unavailable (CERT/CC 2000). The *potential incidents* that could follow masquerading and spoofing are obviously misuse (unauthorised access and alteration of information), malware (introduction of Trojan horses, viruses / worms) and software compromise (Buffer overflow, DoS).

The *severity* of masquerading and spoofing is considered high in external servers, as it may result in loss of availability. The *urgency* to respond is high as well, since IP spoofing can very soon escalate to a DoS incident. However, even in the case of masquerading, once unauthorised access is achieved to external servers, it is possible to alter information that can harm the public image of the organisation and thus cause further embarrassment and disruption of operation.

In the case of internal servers, even if services are not accessed externally, the danger of disclosing confidential information is considerably high, resulting in severe embarrassment to the organisation, and disruption of its operation. So, the level of *severity* and the *urgency* to respond in this case are high as well.

As for user workstations, the *severity* is less significant, as in many cases the nature of information or access level obtained will not pose a great level of threat to the system (although some users will always be exceptions). The level of *urgency* is medium as well, since the workstation is probably used as a step to achieve increased access into a more significant component of the system (either internal or external server).

Obtaining unauthorised access in network components, as well as making them unavailable by achieving DoS attacks, is highly severe, as it can affect multiple hosts or even the entire internal network, depending on the scale of the problem. The *urgency* to respond is thus high as well.

5.3.2.2 Bypass

Bypass is an action taken to avoid the authentication process by using an alternative method to access a target. For example, some operating systems have vulnerabilities that could be exploited by an attacker to gain privileges without actually logging into any privileged account. Bypass is usually a result of software compromise (e.g. buffer overlow) or malware (e.g. if a Trojan horse is used instead of the original authentication process). The issue is again a threat to confidentiality, as increased unauthorised access is achieved. The potential incidents that can follow are misuse (unauthorised access and alteration of information) and malware.

The *severity* is medium in the case of external servers, since their availability is not threatened directly. However a rapid response is needed to avoid further escalation of the incident, so the *urgency* in that case is high. In internal servers both *severity* and *urgency* are high, as the direct threat is higher, so is the need to avoid escalation of the incident. Although the *severity* in the case of user workstations is lower, and thus can be considered as medium, the need to respond is equally high, since bypassing authentication is a strong indication of an already compromised system, so further action should be taken as soon as possible. Finally, bypassing authentication in network components is of medium severity, since the threat to confidentiality is not as severe as in the case of internal servers, but again the need to respond and eliminate any chances of escalating the problem is high.

5.3.3 Software Compromise

Intrusions that involve the exploitation of software vulnerabilities fall into this category. As already discussed in section 2.1.1, there are three main categories of vulnerabilities within a system, namely design, implementation, or configuration vulnerabilities. The main categories of intrusions that fall into this category are Buffer Overflow and Denial of Service and these are presented below.

5.3.3.1 Buffer Overflow

Buffer overflow is a result of deficient software implementation that allows the assignment of data to a buffer without checking in advance if its size is sufficient to 'host' that data. So in the case of someone sending larger amounts of data, the targeted system will allow the input of data in the buffer anyway, with the result of either crashing the system or overwriting part of memory adjacent to the buffer (possibly containing values of the stack pointer, function return addresses, programming code). As a result of the latter, unauthorised access could be obtained by modifying the flow of program execution, and allowing the execution of arbitrary code with the same access rights granted to the compromised program (Aleph1 1996).

Although implementing software with security in mind can easily prevent buffer overflows, such incidents are quite common and can compromise the integrity and availability of the targeted system. Buffer Overflows are more commonly exploited in server software (web, ftp, email, file) since, firstly, they are easily accessible from external sites and, secondly, they usually run under root/administrator privileges. Buffer overflows can lead to further incidents such as bypassing authentication, denial of service, misuse or execution of malware. In all cases, the amount of time elapsing before that happens is usually very small, as in many cases it even happens almost immediately (e.g. as part of a broader automated exploit).

As described earlier, external and partially internal servers are more vulnerable to buffer overflows, having greater risk of allowing root access privileges to unauthorised parties and also disrupting their operation. Thus the *severity* in both cases is high. The *urgency* to respond is high as well, as the likelihood of escalation is significant, leaving the need for an urgent response.

In the case of user workstations the *severity* is medium, since the chance of being subjected to attacks of this nature is less significant. Also, even if targeted (e.g. server software is running, probably by default) the number of hosts affected are limited (probably only one), so the scale of the problem is less significant. However, the *urgency* to respond is still high, in order to avoid execution of malware or further compromise of other systems.

The chance of exploiting buffer overflows in network components is even less common, but the potential impacts of doing so are more serious than in the case of workstations, since a greater number of hosts can be affected (CERT/CC 2002). Thus the *severity* of buffer overflow is medium in this case. The *urgency* to respond is again high, for the same reason.

5.3.3.2 Flood / Denial of Service

'Denial of Service' (DoS) attacks aim to make the target unable to respond to any other events / requests and thus become inaccessible to legitimate clients. In most cases that aim is achieved by overloading (flooding) the capacity of a target after accessing it repeatedly, whereas in other cases the target is confused and freezes, after receiving malformed information (Internet Security Systems 1997). The result of such action is not to break into systems, but make it inaccessible to others.

Some examples of DoS attacks are repeated requests to open connections to a port on a network (SYN flooding), reception of large number of fragmented crafted packets, or initiation of processes on a computer (in order to consume resources). Another example is the reception of high volume of e-mail messages addressed at a single account, which exceeds the resources available. Finally, another example of DoS attack is SMURF attacks, which were already described in section 2.1.1. After the occurrence of a DoS attack against a target, and the success of making it inaccessible, another party could take over the role of the target and act on its behalf, resulting in spoofing. As an example, it is possible to hijack TCP connections of the target (Internet Security Systems 2004) and thus access information without authorisation.

The impact of DoS attacks clearly relates to the availability of the targets. Since these attacks are most often conducted with the use of automated scripts, the need to respond immediately is crucial in most cases.

In the case of an external server, the *severity* is high, given that a public-facing site represents a public interface of the organization, and inaccessibility could result in embarrassment and loss of custom. The *urgency* to respond is also high, since usually the time available to prevent either the occurrence of the incident, or subsequent escalation, is very limited. Although DoS to internal servers and network components does not risk causing direct embarrassment to the organisation, their failure to provide services could have impact on multiple hosts, or even the entire internal network of the organisation, so the *severity* is also high, as is the *urgency* to respond. In the case of user workstations, the likelihood of being subjected to a DoS attack is rather small, simply because the impact of doing so is not as significant. User workstations are mostly used as (potentially unwitting) tools to conduct DoS attacks in order to achieve maximum level of effectiveness, but are not the targets. However, it is possible, and it can result in either degradation of performance, or total loss of legitimate usability. Thus the *severity* in that case is medium. The *urgency* to respond is medium as well, as the impacts of the attack are of medium severity and the time available to encounter the attack or avoid escalation is usually more.

5.3.4 Malware

Malicious software, also known as malware, characterises the classes of intrusions that are conducted under complete software control. Intrusions falling into this category differentiate from automated software tools used to launch other classes of attacks (e.g. DoS attacks), in the sense that humans are not involved in the escalation of malware attacks; after the initial human involvement to begin the distribution of malware, individual attacks can subsequently occur without the need for the instigator's further involvement (Furnell 2001). Thus malware can constitute an attack in its own right. There are three main types of malware, namely Trojan horses, viruses, and worms, and all are discussed below.

Trojan horses take their name from the hollow wooden horse that the Greeks used to invade Troy, by misleading Trojans to accept it in their city as a harmless gift, whereas in fact, it had been used to conceal Greek soldiers inside it (Homeros 850 BC). Similarly, Trojan Horses are programs that appear to perform a useful or harmless function, while they actually contain hidden functionality that is unknown to the user. This functionality is intentionally implemented and will typically cause unwanted and often damaging effects for the unsuspecting user.

According to Brunnstein, a virus is a non-autonomous set of routines that can replicate itself, by modifying programs or systems in order to contain executable copies of itself. A worm on the other hand is... "a set of programs or routines that are capable of independently, or with the help of an unsuspecting user, propagating throughout a network" (Brunnstein et. al 1990). Both viruses and worms have the ability to carry malicious code as payload that can result in compromising a system. The main difference between them is the way they replicate themselves; viruses need to infect some host (e.g. file or system boot sector) in order to be activated, while worms are autonomous programs that do not need to infect other programs in order to replicate themselves and get activated. Also, the replication of a worm can quite often result in significant consumption of both computer memory and network resources, thereby leading to a degradation of performance (Furnell 2001). An example of such a case is the Slammer worm, already discussed in section 4.1.

The impacts of malware can differ significantly from case to case, since the code in the payload can do nearly everything that is feasible under software control. For example, it is possible to initiate posting of legitimate users' working documents to all the members of his address book, resulting to breach of confidentiality (CERT/CC 1999). Alternatively, it is possible to delete or modify files in the system, achieving a breach of integrity. Finally system or network resources can be consumed at the execution of the payload, resulting in either degradation of performance or entire inaccessibility of targets for legitimate use.

The potential incidents that can follow the execution of malware can also be nearly anything. Misuse, other forms of malware, software compromise and information gathering are examples of potential results of malware. Thus the severity of malware varies according to the specific incidents. However, if considering the execution of malware in general, the *severity* is high in all types of targets, since such a great variety of functionality can potentially be included in the payload. In addition, the risk of spreading to additional targets is extremely high, so the *urgency* to respond and contain the execution of malware is high as well in all cases.

5.3.5 Misuse

Misuse relates to unauthorised or unacceptable use of system resources. In this sense, it is a quite general term that can actually include all the incidents described so far, since all of them are somehow a form of misusing system resources. However, incidents falling into this category mainly take place after unauthorised access has been obtained in a target and include cases that mainly involve misuse of files and data within a system. It is important to mention at this point that the occurrence of incidents from this category indicates that the targeted system may have already been in a compromised state, unless the IDS is wrong and the activity is legitimate and perpetrated by a legitimate user.

5.3.5.1 Unauthorised Alteration

Unauthorised alteration includes actions such as creating, modifying, deleting system or data files. This will affect the integrity and / or availability of resources, and represents an important issue that needs to be addressed.

The *severity* in the case of external servers is high, as information or services might be altered in such a way as to cause embarrassment to an organisation and further disruption of its normal operation. For example, web site defacements (Zone-h 2004) represent a highly important incident that can immediately attract the interest of media and put the organisation into a difficult situation. In addition, the modification of information or services could potentially mislead or cheat customers, and result in making the organisation liable for those actions. Although the *urgency* to respond in such case is high, the feasibility of doing so might be another issue. Certainly the current state of the system needs to be considered in order to determine the effectiveness or selection of an appropriate response.

Unauthorised alteration is highly severe in the case of internal servers and network components as well, since it can result in misleading internal users to make decisions based upon inaccurate information or disrupting their operation. Even if the likelihood for rapid escalation of the incident is very small, the need for timely response is high again, since the severity of the incident can be so significant.

Finally in the case of user workstations, the importance of the target is typically lower, as it can affect only a limited number of users. The *severity* is therefore medium. Still, the *urgency* to respond is high, mainly because the current state of the targeted system should be assessed and any potential risks minimised.

5.3.5.2 Unauthorised Access

Unauthorised access includes actions that involve disclosure of information to unauthorised parties. Incidents from this category can breach the confidentiality of a system. As a result of their occurrence, incidents such as unauthorised alteration or execution of malware might follow. Thus the *severity* of unauthorised access can vary according to the target (and whether confidentiality is at high risk) but the *urgency* to respond in all cases should be high. That is because the current state of the system should be assessed, and further escalation of the incident prevented (e.g. occurrence of unauthorised alteration or execution of malware).

When external servers or network components are subjected to unauthorised access, the severity is low, since no confidential information should be at risk and no modification has taken place. On the other hand, the current state of the system is unknown and needs to be assessed. By contrast, unauthorised access to internal servers has high severity, because there is more important information available for attackers. In the case of user workstations the severity is medium, as there is risk to confidentiality, but it is less substantial.

5.4 Conclusions

In this taxonomy, several categories of incidents have been considered, aiming to illustrate the effect of different types of targets on the results of an intrusion. The ultimate intention is to give insight into the main intrusion characteristics (Table 5.1) that can influence intrusion response, and subsequently lead to the indication of generic classes of response. Although the response-oriented taxonomy is quite generic, and cannot depict the complexity of the response decision process, it still serves as a basic tool that enables the research to progress towards that direction. After examining the results of different intrusions on various targets, it becomes apparent that intrusions directed towards internal servers always have the most significant results, mainly due to their importance in the operation of an organisation. By contrast, user workstations have the least significant results, as their role within the organisation is less important and the consequences after the occurrence of an intrusion can more easily be addressed. Finally, network components and external servers seem to depend upon the type of intrusion to a greater extent, as some classes of intrusions have more significant effect than others.

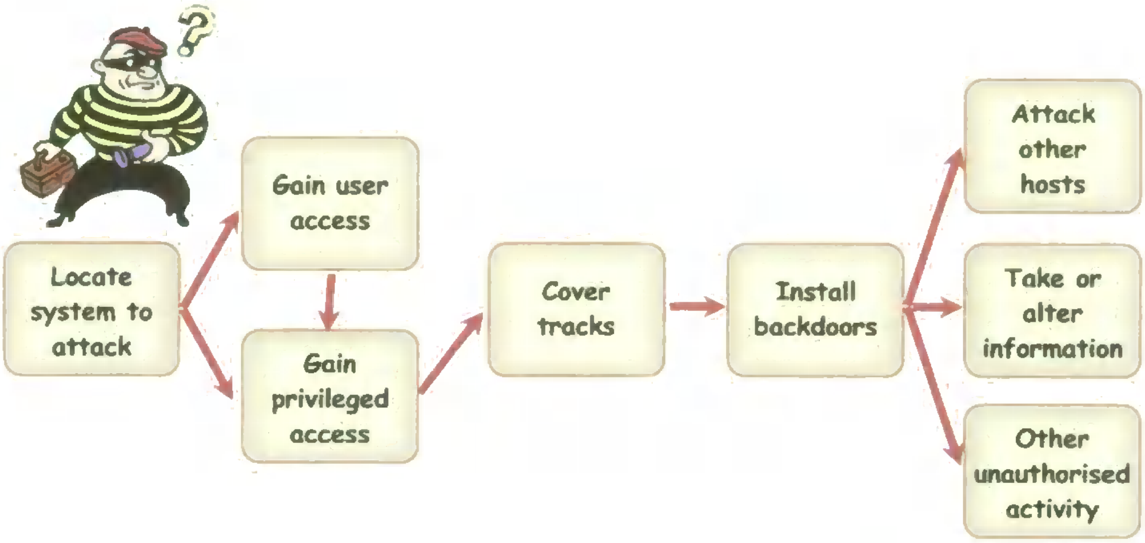
In terms of response, and how different intrusion characteristics can influence the response process, it can be argued that the more severe an intrusion is, the more important it is for the response to focus upon the prevention of its occurrence, and / or its containment. In classes of intrusions with high urgency, the risk of rapid escalation is significant, and so the response process should focus upon the prevention of further escalation (i.e. preventing the occurrence of the potential incidents). Finally, the severity and urgency can affect the intrusiveness of the initiated response. It is apparent that there should be a trade-off between them, as the more severe the intrusions, the more intrusive responses can be applied.

Also, it is possible to distinguish the different phases of attacks and the different ways a response mechanism should counter each one of them. *Information gathering* attacks aim to facilitate the process of compromising systems, and although they generally do not cause a great level of disruption to organisations, they enable attackers to locate which systems are vulnerable and can be attacked. Even though attackers could still blindly attack systems, in the hope that some of them will prove to be vulnerable, having intelligence about potential targets can definitely make their task more effective. Responses to *information gathering* attacks should aim to prevent attackers from getting any useful information, and at the same time, make sure that any vulnerable systems are patched. Admittedly, that is easier said than done, since the task of continuously updating vulnerable systems could prove to be overwhelming for the IDS. Also, preventing attackers from getting information could be achieved by either denying / stopping their requests, or using deception to provide false information. However, in the event of a legitimate request, the IDS would then be doing more harm than good.

In order to compromise a system, it is important to obtain access to it first (with the exception of DoS attacks, which do not require access). Thus, the next phase of an attack would be used to provide illegal access to a system. *Authentication failure*, *software compromise* and *malware* attacks can be used for that purpose. Responding to such attacks should involve preventing them from happening, when possible (i.e. when there is very low probability of a false alarm) or containing them as much as possible, in order to avoid their escalation. That could involve increasing the monitoring level, which could determine whether the activity is malicious or not, and as soon as an attack is detected, then the focus should change to eliminating the attack, and restoring the system to its initial state, as much as possible.

Once access has been obtained in a system, the attacker is able to abuse it, by performing *misuse*, or *malware* attacks. Also, the system could be used as a stepping-stone to attack other systems and perform a whole new cycle of attacks against different targets. Responding to these attacks should involve eliminating them and restoring the system, to its initial state, as much as possible. At the same time, it is important to prevent the attacker from targeting other systems in the organisation. CERT/CC illustrates the process of a typical network attack in Figure 5.1. One final remark is that the certainty of the IDS about the occurrence of an actual attack, instead of a false alarm, should increase, as the stages of an attack progress, and thus the severity and transparency of the responses it issues should be adjusted to account for that fact. Determining the probability of a false alarm will be discussed in Chapter 6.

It should be noted that there are several limitations in this taxonomy. For example, apart from the type of target, the number of systems targeted could also be considered, as the scale of an incident will certainly influence its severity. However, the omission of this factor does not prevent the taxonomy from fulfilling its previously stated objective of demonstrating that the same category of incident can demand different responses in different contexts.



(source: CERT/CC 2003b)

Figure 5.1 Phases of a typical network attack

Finally, it should be noted that the taxonomy is intended to provide the foundation for an automated decision mechanism within a response software agent. However, it is evident that the decisions of the response mechanism should be more elaborate than the generic recommendations presented in this section, and should depend upon a greater variety of contextual factors. Although incident and target related characteristics are the main determinant of the likely result of the incident, various other contextual factors could be measured, when an incident is detected, in order to better inform the response decision process. For example, the decision capability of the response mechanism, the probability of a false alarm, the user account in use, the current alert level of the IDS, and the nature of any responses already issued could all influence the choice of response that is likely to be the most effective. Further consideration of this issue is presented in the next chapter.

CHAPTER 6

A CONCEPTUAL ARCHITECTURE FOR A FLEXIBLE AUTOMATED INTELLIGENT RESPONDER

6 A CONCEPTUAL ARCHITECTURE FOR A FLEXIBLE AUTOMATED INTELLIGENT RESPONDER

6.1 Introduction

In the effort to increase the awareness of a response system at the time of an attack, and consequently provide the basis for it to operate autonomously, the Response-Oriented Taxonomy of Intrusions was developed. After systematically studying intrusions, the top-level factors influencing intrusion response were identified. These are depicted in Figure 6.1, according to whether they are related to the incident, or the IDS.

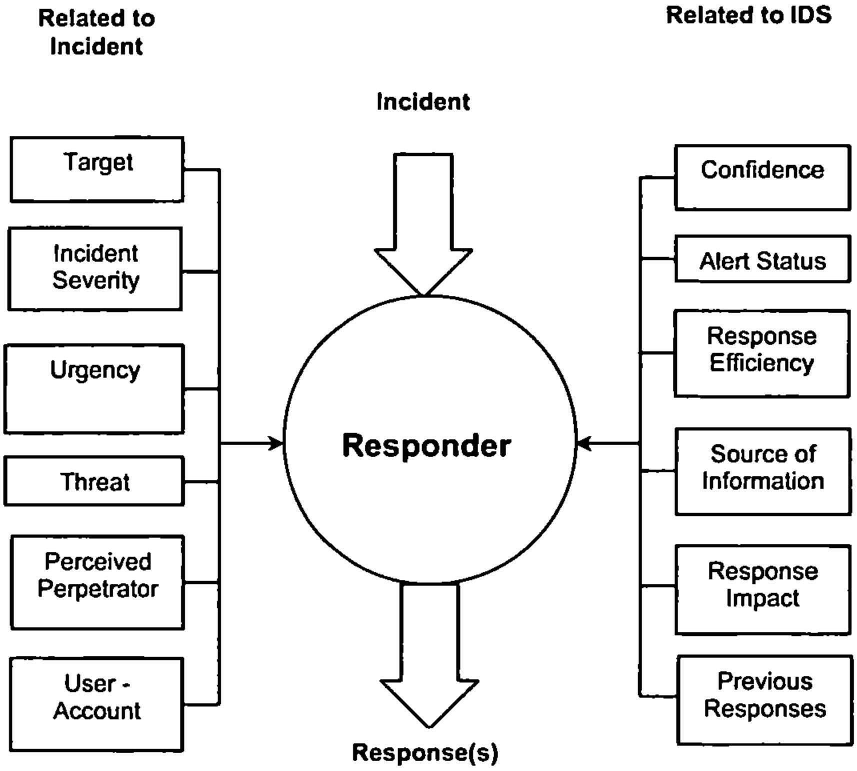


Figure 6.1 Main contextual factors influencing intrusion response

As Figure 6.1 shows, the incident is the trigger for the response and still represents the principal influence over what should be done. However, assessment of the other influencing factors enables the responder to establish the context in which the incident has occurred, and therefore select appropriate responses accordingly. Some of the factors that are related to the Incident can be directly linked to the intrusion characteristics covered in the Response-Oriented Taxonomy (chapter 5). For example, the Target relates to the 4 types of targets covered in section 5.3, and the Incident Severity, Urgency can be directly linked to the Severity and Urgency characteristics of section 5.3. The various factors related to the incident are defined in more detail as follows:

- **Target**: what system, resource or data appears to be the focus of the attack? What assets are at risk if the incident continues or is able to be repeated? How important is that resource for the continuation of the system operation?

- **Incident severity**: what impact has the incident already had upon the confidentiality, integrity or availability of the system and its data? How strong a response is required at this stage? For example, the detection of a severe incident could warrant the initiation of correspondingly severe responses, in order to protect system resources.

- **Urgency**: How urgently is a response needed? This factor will be influenced by several of the other factors.

- **Threat posed by incident**: how serious is the threat to the system, after the occurrence of the incident? Which attacks are more likely to follow, after that incident?

- **Perceived perpetrator**: does the evidence collected suggest that the perpetrator is an external party or an insider? Is there any history associated with that person/account?

- **User account**: if the attack is being conducted through the suspected compromise of a user account, what privileges are associated with that account? What risk do those privileges pose to the system?

In addition, the factors related to the IDS are summarised as below:

- **Confidence**: how many monitored characteristics within the system are suggestive of an intrusion having occurred?

- **Alert status**: what is the current status of the monitoring system, both on the suspect account / process and in the system overall?

- **Response efficiency**: what has the efficiency of a specific response proven to be under specific conditions? The IDS will gradually update the efficiency rating of a specific response, after considering its efficiency in previous incidents. For example, for some types of attacks, targets, or attackers, some responses might be more efficient than others.

- **Source of Information**: what is the detecting capability of the source of information about the incident? Some sources or IDS metrics might be more reliable in detecting attacks than others, generating less false positive alarms (e.g. anomaly detectors tend to generate more false positive alarms than misuse detectors (Bace and Mell 2001), and some monitoring sensors produce fewer false alarms than others, depending on their location and configuration). The IDS should be able to determine the credibility of sources over time and adjust the confidence of the system on the probability of an intrusion.

- **Response impact**: what would be the impact of initiating a particular form of response? How would it affect a legitimate user if the suspected intrusion were, in fact a false alarm? Would there be any adverse impacts upon other system users if a particular response action were taken? Would it be possible to eliminate any adverse impacts and return the system to its initial state?

- **Previous Responses**: have any responses already been issued as a result of this incident? If one or more responses have already been issued, and been unsuccessful in countering the intrusion, it would be relevant to consider this before determining the acceptable impact of the next action. The failure of previously issued responses might lead to the selection of more severe response actions (or an increase of the overall alert status of the system).

Having identified these factors, it is necessary to consider a response architecture within which they can be used. As such, the conceptual architecture for a Flexible Automated Intelligent Responder (FAIR) is proposed. The next section presents an overview of the FAIR architecture. The discussion then proceeds to consider the operational characteristics of FAIR, as a novel approach to the problem of automated response. Then, the main modules of FAIR are described, focusing upon the contextual factors they provide, or assess, and their role in the intrusion response process. Finally, the Response Policy is presented, describing the response decision process in more detail.

6.2 Flexible Automated Intelligent Responder (FAIR)

FAIR has been based upon the Intrusion Monitoring System (IMS), a conceptual architecture for intrusion monitoring and activity supervision, focused around the concept of a centralised host handling the monitoring of a number of networked client systems. Intrusion detection is based upon the comparison of current user activity against both historical profiles of normal behaviour for legitimate users and intrusion specifications of recognised attack patterns (Furnell 1995). The original IMS specification provided no detail regarding the design and operation of the Responder entity, and thus FAIR represents completely new work in this respect.

The reasons for using IMS as the underlying detection system relate to its concept of performing both host-based and network-based intrusion detection. Also, the simple host-client model used in the IMS architecture lends itself to demonstrating the concepts of intrusion detection and response more easily, without adding additional levels of complexity that relate to distributed architectures. In a way, this can be considered as a limitation of this research, in the sense that its simple architectural model would need to be modified, if the FAIR System was to be applied in large network environments. However, proving the viability of the FAIR System and its concepts is the focus of this research, and adapting these concepts to more complex architectures represents an area in which the research can be extended in the future.

FAIR uses an expert system. The expert system technology was selected due to its ability to represent uncertainty, which is very important for the representation of complex problems (Giarratano and Riley 1998). Although conventional programming languages have the potential to model uncertainty and abstraction, considerable programming effort would be required to achieve the same level of inference on knowledge, as expert systems, which are specifically designed for this task. Other potential technologies, such as fuzzy logic were considered, but they did not offer the same level of documentation, support and integration with Visual Basic, as CLIPS (C Language Integrated Production System) did. Finally, since machine learning is not supported by the FAIR architecture at this stage, neural networks would not add much value to the decision engine, and thus they were not chosen for the development of the response decision engine. The elements of FAIR are illustrated in Figure 6.2, and discussed below.

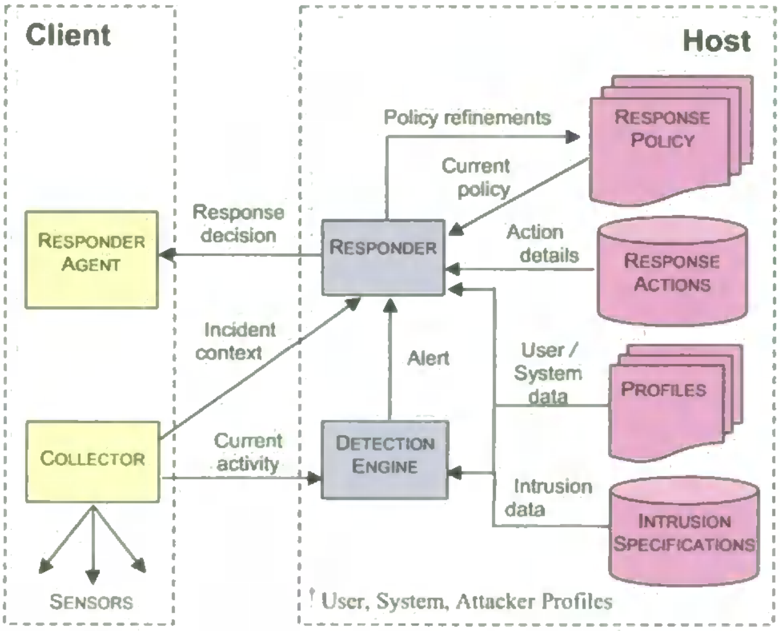


Figure 6.2 The FAIR Architecture

- **Detection Engine**: As well as indicating the name of the suspected intrusion, the *Detection Engine* can directly inform the *Responder* about a variety of other factors, including the target of the attack, and the perceived perpetrator.

- **Responder**: The *Responder* is responsible for monitoring the Alerts sent from the *Detection Engine* (note: this module was referred to as the Anomaly Detector in the original IMS Architecture). After considering alerts, in conjunction with other contextual factors, it takes appropriate actions where necessary. In order to reach a decision, the *Responder* retrieves a variety of information, which is acquired from the *Detection Engine*, the *Intrusion Specifications*, the *Profiles*, the *Response Actions*, and the *Collector*.

- **Intrusion Specifications**: *Intrusion Specifications* contain information about specific types of intrusions and their characteristics, such as incident severity rating, ratings of likely impacts (e.g. in terms of confidentiality, integrity and availability), and the speed with which the attack is likely to evolve. Once the *Detection Engine* has indicated the name of intrusion that it believes to have occurred, additional information can be retrieved from the specifications to obtain a comprehensive view of the incident.

- **Profiles**: The *Profiles* contain information about users, systems, and attackers, all of which can provide some information in the context of response decisions:

  - **User profiles**: If the incident involves the utilisation of a user account, then the corresponding user profile can indicate aspects such as the privileges and access rights associated with it.

  - **System profiles**: These relate to system characteristics, which enable the Responder to get a clearer picture of the target, and ensure that important services, or information can be protected as much as possible.

- **Attacker profiles**: These relate to information about known attackers, as they have been collected either from forensic analysis during previous incident investigations, or after research about attacker profiles (Honeynet Project 2003). Obtaining information about attackers is understandably not a trivial task, and there are many challenges associated with it. Still, the contribution of such information for the enhancement of intrusion response is very significant.

- **Response Actions**: These relate to the characteristics of response actions available within the FAIR System. They are used to select the responses with the most appropriate characteristics.

- **Collector**: After the *Responder* receives an alert, it can communicate with the local *Collector*, to request information about current activity on the target system (e.g. applications currently running, network connections currently active, applications installed, load of the target at the time, etc.).

- **Response Policy**: Having gathered all of the available information, the actions that should be initiated in different contexts are then specified in the *Response Policy*. Specifically, the *Response Policy* uses expert systems technology to indicate the most desirable characteristics the selected responses should have under the circumstances, and then estimate how closely the available response actions match those characteristics in order to select the stronger matches.

- **Responder Agent**: If the actions selected by the *Responder* need to be performed on the client side (target), a local *Responder Agent* is responsible for initiating and managing the process. Without providing an exhaustive list, examples of actions that could be performed at the client side include correcting vulnerabilities, updating software, issuing authentication challenges, limiting access rights, and increasing the monitoring level.

6.3 Operational Characteristics of FAIR

In order to provide the foundation for automated response, two characteristics are considered to be very important. The first is the ability to represent the context of an attack and thus enable a Responder to make as informed decisions as possible. The second characteristic is the ability to operate with great flexibility, and reflect, as a result, the changing characteristics of organisational networks. FAIR incorporates techniques to acquire these characteristics, and they are discussed in the sub-sections that follow.

6.3.1 Adapt decisions according to the ability to make the right choices

One important aspect is the adaptation according to the ability of the Responder to make correct decisions. Indeed, the first basic step of enabling the Responder to operate autonomously is to assess its ability to make appropriate decisions, and then adapt the level of impact it can have upon systems and users accordingly. The decision making ability of the Responder can be determined after considering which decisions were correct and which were not. As the percentage of correct decisions grows, the Responder will be able to play a more significant role in the protection of systems, and data, by being able to issue increasingly severe responses.

6.3.2 Assess the appropriateness of responses before and after initiating them

Another novel feature of the Responder is the way the appropriateness of response actions is assessed. So far, cost-based models have been used to determine and balance the cost (impact) of a response, against the cost (impact) of the impending attack (section 4.4.7). The FAIR architecture uses the same principle, but moves one step further by offering greater flexibility in the decision criteria, and introducing a more detailed (and accurate) representation of the response and attack impacts. In addition, it enables the use of a feedback mechanism, which assesses these impacts after the responses are issued, in order to provide the basis for improvement. Specifically, the main considerations in selecting a response are based upon its potential side effects, and its practical effectiveness in fulfilling its intended role.

As previously identified, the problem of side effects is a particular concern in the context of using active responses, because they have the potential to adversely affect legitimate users of the system. As a result, this needs to be considered before the Responder chooses to initiate a given action. There are a number of characteristics that would be relevant in this context:

- the transparency of the response action. In some cases it might be preferable to issue responses that do not alert the attacker to the fact that he has been noticed, whereas in others it could be preferable to issue a response that is explicit.

- the degree to which the action would disrupt the user(s) to whom it is issued. This is especially relevant in the context of a response action having been mistakenly issued against a legitimate user instead of an attacker. In situations where the

Detection Engine has flagged an incident but indicated low confidence, it would be desirable to begin by issuing responses that a legitimate user would be able to overcome easily.

The practical effectiveness of the response in fulfilling its role can be reflected in its efficiency. Assessing the efficiency of a response involves the assessment of its appropriateness, after the response is initiated. This requires some form of feedback, which could be provided in two ways: explicitly by a system administrator, and implicitly by the Responder. In the former case, the administrator would inspect the alert history and manually provide feedback in relation to the responses that had been selected to indicate whether or not they had been effective or appropriate to the incident. In case of badly issued responses, the administrator would be able to inform the system whether the response was too severe, whether it was not severe enough, whether it had unwanted side effects, whether it was applied too late, whether it was completely inappropriate, or whether it was applied in a false alarm scenario. This information can be used to adjust the effectiveness of the response action and the Responder.

By contrast, the implicit feedback would require the Responder to infer whether previous responses had been effective. A simplified example of how it might do this would be to determine whether it had been required to issue repeated responses in relation to the same detected incident. If this was the case, then it could potentially infer that (a) the initial response actions were not effective against that type of incident, and (b) the last response action issued might form a better starting point on future occasions (i.e. upgrading and downgrading the perceived effectiveness of the responses when used in that context). Another example would be to incorporate mechanisms that can determine the result of an attack (whether the attack has been successful), and in this way, indirectly assess the effectiveness of the responses issued against it. For example, referring back to the typical phases of an attack in section 5.4, if a Trojan horse is installed after a buffer overflow attack, and is part of the same intrusive activity, then the Responder can infer that the responses issued for the buffer overflow attack were unsuccessful, as they allowed the incident to progress and escalate. The feasibility of such a feature largely depends upon the ability to determine the result of an attack, and although such a feature (attack verification) is not mature yet, initial work has already started emerging (as discussed in sections 3.3.2, 3.3.6, and 3.3.7).

Having obtained such feedback, it would be desirable for the system to automatically incorporate it into a refined version of the Response Policy. This, however, would be a non-trivial undertaking, and it is anticipated that a full implementation of the system would need to incorporate machine-learning mechanisms to facilitate a fully automated process (Mitchell 1997). An alternative would be to collate the feedback, and present it to the system administrator for later consideration when performing a manual overhaul of the Response Policy.

6.3.3 Adapt decisions to account for changes in the environment

A fundamental principle, which has been emphasised before, is that response decisions should vary depending upon the context in which the incident has occurred (i.e. a response that is appropriate to a particular type of incident on one occasion will not necessarily be appropriate if the same incident was to occur again under different circumstances). In order to do that, it is important to account for changes in the environment and adapt response decisions accordingly. In the Automated Intrusion Response Model (section 4.4.5), the impact of responses is dynamically assessed to account for changes in affected systems and services at the time of the incident. Although the only changes accounted for were results of response actions, such as disabling or limiting access to specific services, it is still an important feature that increases the awareness of the responder system. This approach, though, could be further extended to reflect more aspects of the environment. When the Responder draws upon information from a number of other sources within the FAIR system, it enables the assessment of the overall context in which an incident has occurred, including considerations such as:

- the overall alert status of the IDS at the time of the new incident;

- whether the incident is part of an ongoing series of attacks (e.g. how many targets have already been affected? Which responses have already been issued? How dynamic is the occurrence of new incidents?);

- the current status of the target (e.g. is it a business critical system? What is its load at the time? Are there any active users connected? Is there any important information or service that needs to be protected? Is there any software (e.g. auditing, or highly vulnerable software) running at the target that can introduce additional risk? What software/hardware can be used for response?);

- the privileges of the user account involved (e.g. what is the risk of damage to the system?);

- the probability of a false alarm (how reliable has the sensor/source that detected the incident been in the past? What is the level of confidence indicated by the Detection Engine about the occurrence of an intrusion?);

- the perpetrator of the attack (is there enough information to suggest a specific attacker? Is he an insider/outsider? Has he initiated an attack before? How dangerous is he? What attacks is he likely to attempt?);

Having assessed the above factors, response decisions must then be adapted to the context accordingly.

6.3.4 Offer flexible and escalating levels of response

Another feature of FAIR is its ability to offer escalating levels of response to account for the varying levels of threat introduced by incidents. A similar feature was adopted in the Automated Response Broker (ARB), in which escalating levels of response were offered, according to the risk introduced by the intrusion and the effect of the response (section 4.4.4). The main advantage of this approach is that it overcomes the problem of existing IDS/IPS solutions, which can either permit or deny a security event, and do not offer any scalability, which is essential for cases of false positive alarms. Moving one step further from ARB, the FAIR approach supports a wider variety of response actions (other than stop the attack and restore the system), enabling a greater level of flexibility for a variety of events. For example, in the event of a suspicious user login, the Responder can select responses with a variety of severity levels, according to the circumstances. Possible options would include just logging the event and doing nothing else, alerting the administrator, allowing the user to login but increasing the monitoring with keystroke analysis, allowing the login but limiting access rights to prevent potential damage to the system, issuing an explicit authentication request in the form of a password or associative question before allowing the login, or denying access to the user altogether.

The basis for achieving scalable responses is the assessment of the overall threat introduced to the system after the occurrence of an attack, and the estimation of the impact of a response action. Finally, a great level of scalability and flexibility is achieved by providing a user-friendly interface, which is used to facilitate the customisation of Response Policies.

6.4 FAIR Modules

This section discusses the main modules of the FAIR architecture, focusing upon the contextual factors they consider, and their role in the intrusion response process. The process of determining these factors was based upon studying case studies of intrusions and reviewing literature about general security, computer crime, forensic analysis, and manual incident response (Mandia and Prosise 2001). The factors are illustrated in Figure 6.3, and their description will focus upon the following issues:

- **How each response factor can be assessed within a system**

  Some factors are pre-defined by the system administrator; others are assessed dynamically by the Detection Engine, the Responder, or the Collector, whereas others are refined gradually over time, based upon historical events.

- **How their values can influence the decision process of the Responder**

  The process of decision-making involves a combination of different weightings for factors, and determination of specific thresholds that need to be exceeded to enable the initiation of specific responses. It is important to comment at this stage that the values of weightings and thresholds used in this architecture are only indicative, as the establishment of meaningful values would involve further research in its own right, with the use of real incident scenarios as case studies, and possibly the collaboration with Incident Report Teams, such as CERT/CC, Symantec Response Centre, and so on. Thus, although the logic behind selecting specific values can be justified, focus of this research is not to provide a fully accurate response mechanism.

The values for many of the factors depicted in Figure 6.3 (e.g. Intrusion Confidence, Direct Impact, Overall Threat) are represented on a scale. However, the factors are not all rated equally, and the following conventions have been used for setting the appropriate scales:

- Factors relating to confidence metrics have been represented on a 100-point scale.

- Primary factors that are not computed by the Responder have been represented on a 10-point scale. The choice of a 10-point scale was based on conventional risk analysis methodologies for factors such as Direct/Potential Impact, Severity, and Threat. Then, the need to apply a common scale among all primary factors, in order to facilitate the computing of other factors, lead to the adoption of a 10-point scale for all of them.

- Factors computed by the Responder, which receive input from other factors, have been represented on a 100-point scale. These factors are based on average values, and thus have greater potential for variability. As a result, they require a greater scale.

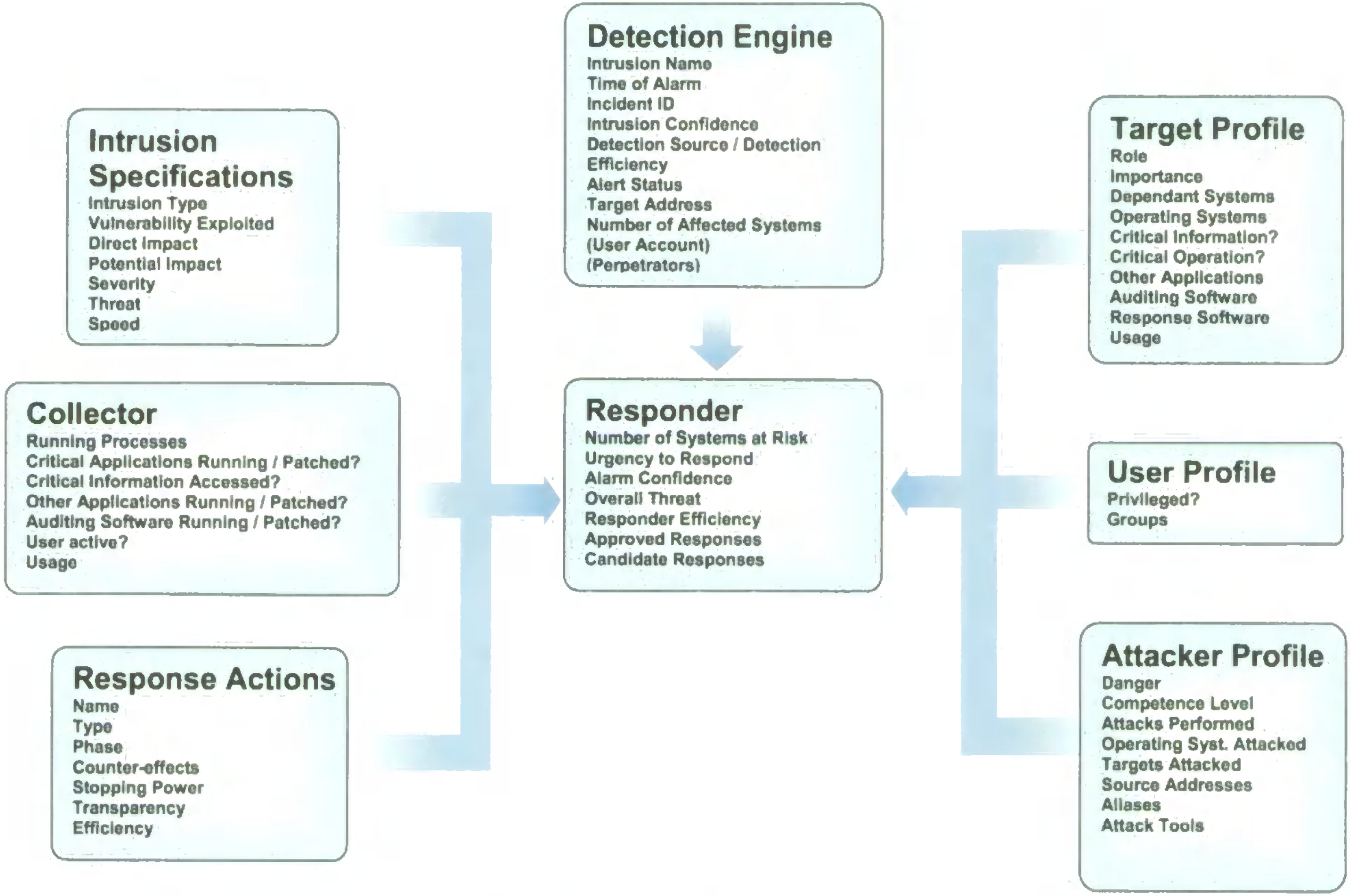
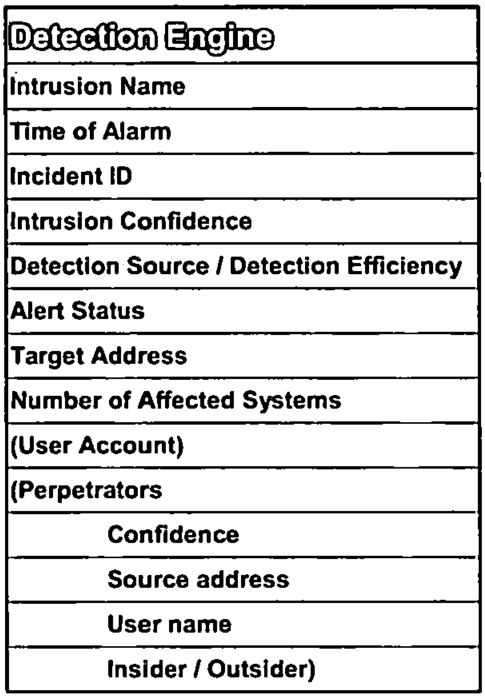


Figure 6.3 Summarised Response Factors

6.4.1 Detection Engine

As already discussed, the *Detection Engine* directly informs the *Responder* about a variety of factors, which are listed in Table 6.1 and described below.



Detection Engine
Intrusion Name
Time of Alarm
Incident ID
Intrusion Confidence
Detection Source / Detection Efficiency
Alert Status
Target Address
Number of Affected Systems
(User Account)
(Perpetrators
Confidence
Source address
User name
Insider / Outsider)

Table 6.1 Response Factors provided by the Detection Engine

6.4.1.1 Intrusion Name and Time of Alarm

These factors represent the name of the suspected intrusion, and the time the alarm was generated. Both factors are determined dynamically and passed on to the Responder as part of the IDS alarm.

The *Intrusion Name* is used by the Responder to retrieve the intrusion characteristics, and the *Time of the Alarm* assists the Responder in estimating the timeframe within which, it needs to respond. By considering that time starts counting down, when the alarm is generated (T_A), a simplistic way of estimating the remaining time to respond (T_R) would be:

$$T_R = U - (\text{Present} - T_A) \qquad (1)$$

The symbol U relates to the Urgency to respond and might represent a timeframe of minutes, hours, or days.

6.4.1.2 Incident ID

The Incident ID serves to uniquely identify the malicious activity that the specific alert is related to. If the alert is a continuation of an existing incident, then the Detection Engine will assign the same Incident ID to all the alerts related to that incident. If the alert is considered to be a new attack, then a new Incident ID will be assigned. Some criteria for relating alerts with each other would be: whether they originate from the same source; whether they involve the use of the same (or related) user accounts; whether they target the same (or related) systems; whether they follow the same patterns of activity; or whether they are a logic continuation of one another (e.g. events that follow the phases of a typical attack, or malware attacks whose methodology is already known).

Correlating alerts is not a new concept, and similar mechanisms have been adopted in many IDSs, including EMERALD and AAIRS (sections 4.4.1 and 4.4.6 respectively). EMERALD performs alert correlation at different levels via the Monitors (*Service*, *Domain*, or *Enterprise-layer Monitors).* In AAIRS, the role of alert correlation is given to the Master Analysis Agent, which creates a new Analysis Agent for each new incident (Carver et. al. 2001). The focus of this research is not so much the alert correlation aspect, which is an issue for research in its own right, but its outcome and its role in the intrusion response process.

Indeed, by associating different alerts with the same Incident ID, the Responder can retrieve the Response actions issued already for the previous alerts, and apart from adjusting the effectiveness of the ones which proved unsuccessful (section 6.3.2), it can escalate the level of severity and urgency with which it needs to respond (section 6.3.4).

6.4.1.3 Intrusion Confidence

This represents the confidence of the Detection Engine about the occurrence of an intrusion, and denotes the significance of the various monitored characteristics that suggest the occurrence of an intrusion. The Intrusion confidence metric is assessed by the Detection Engine, and passed on to the Responder as part of the Alert. It is used by the Responder to determine the Alarm Confidence.

6.4.1.4 Detection Source and Detection Efficiency

As discussed in Chapter 2, the detection capability of IDS systems is not perfect. For example, anomaly detectors tend to generate more false positive alarms than misuse detectors (section 2.3.2.3). Similarly, it is possible that different Collectors (sensors) can provide more credible alarms than others, depending upon their location and the type of events or characteristics they can monitor. Also, in cases of networks with heterogeneous IDS detectors and sensors, where they exchange alarm information, the *Detection Source* and *Detection Efficiency* could serve as a means of assessing the credibility of those alarms. With this in mind, the *Detection Source* denotes the component from which the alarm originates, and the *Detection Efficiency* reflects the credibility of the Detection Source (Collector, Detection Engine), based upon its historical performance.

The Detection Efficiency can be described as the percentage of alarms that correspond to real intrusions, and it is related to the False Alarm Rate (or False Positive Rate) and True

Positive Rate. Both those metrics are widely used in the performance evaluation of intrusion detection systems (Allen et. al. 2000). The False Alarm Rate represents the probability of having an alarm, but not an intrusion, and according to the results of the IDS Evaluation Project from MIT it generally ranges at 0.1%; that is, for every 1,000 sessions of normal activity (legitimate and malicious), the IDS will mistakenly identify one of them as an attack (MIT Lincoln Laboratory 2001). The same study suggests that the percentage of malicious sessions in normal activity is usually 0.001% (one attack every 100,000 sessions). If the True Positive Rate, which reflects the probability of an attack being accurately recognised, is 100%, then the Detection Efficiency is around 1%; that is 1 in around 101 alerts will correspond to a real attack. Axelsson (1999) explores this issue in more detail, as he discusses the impact of false alarms on the performance of intrusion detection systems. In the FAIR system, the calculation of the Detection Efficiency is based upon the average of true alarms over the total number of alarms (false and true alarms). According to the definitions of the True Positive Rate and False Alarm Rate (Axelsson 1999), the Detection Efficiency can be calculated as follows:

$$DE = \frac{\frac{TPR * 0.001}{100}}{FAR + (\frac{TPR * 0.001}{100})} * 100 \therefore DE = \frac{TPR * 0.001}{FAR + (TPR * 0.00001)} \qquad (2)$$

FAR, TPR, and DE represent the False Alarm Rate, True Positive Rate, and Detection Efficiency respectively. The factor 0.001 represents the percentage of malicious activity in normal traffic. The symbols *, and ∴ represent the multiplication sign and the therefore logical symbol respectively.

The False Alarm Rate can be predefined for each detection source, as part of the IDS specification. However, it would be more accurate if refined over time to reflect the efficiency of Collectors at different locations, and the inclusion of new intrusion signatures. The refinement of the Detection Efficiency would be determined by calculating the percentage of true alarms, as shown in Equation (3), where TA, FA and DE represent the number of true alarms, the number of false alarms, and the Detection Efficiency respectively.

$$DE = \frac{TA}{TA + FA} * 100 \qquad (3)$$

Alternatively, and especially for cases of generally low efficiency rates throughout the network, the Detection Efficiency could be adjusted to account for low performance. The Detection Efficiency rates could then be adjusted to reflect only their relative differences in performance, according to their distribution of values. The detection efficiency of sources performing an average rate would be elevated to 50%, and so on.

The Detection Efficiency can be assessed by the Detection Engine, and passed onto the Responder as part of the Alert data. Finally, it can be used to calculate the Alarm Confidence.

6.4.1.5 Alert Status

The Alert Status is assessed by combining all the individual threat levels, as introduced by security events in the organisation, within a specified timeframe. The more severe those events are, the higher the Alert Status will be. In a way, we could say that the Alert Status is a similar metric to the Overall Threat (which will be discussed in section 6.4.6.4), with the main difference between them being that the Overall Threat represents the danger that arises after the occurrence of a specific event, while the Alert Status denotes the danger associated to the system in general, after considering previous incidents as well. So, one could conclude that the Alert Status is a more generic metric, that reflects the security threat present in the organisation and monitored systems at a given moment. A simplified calculation of the Alert Status is described in Equation (4), where AS, OThr, and n represent the Alert Status, the Overall Threat for each event, and the number of events within a specified timeframe. The Alert Status is assessed gradually by the Detection Engine, based upon historical events.

$$AS = \frac{\sum_{i=1}^{n} OThr_i}{n} \tag{4}$$

According to the Alert Status, the Responder can select the appropriate characteristics of an appropriate response, including its severity. For example, the occurrence of the same incident might justify the use of more severe actions, when the Alert Status is already high.

6.4.1.6 *Target Address*

This factor contains the network address of the attacked target. It is determined by the Detection Engine and passed on to the Responder, as part of the Alert. Based upon that information, the Responder can extract information from the Systems Profiles, about target characteristics, and communicate with the Collector at the target, to extract more information about the target and the incident.

6.4.1.7 *Number of affected systems (NAS)*

This metric aims to reflect the extent of the problem that has been caused by an ongoing attack, and reflects the number of systems that have already been affected by the attack. If, for example, the detected attack were a worm, the number of affected systems would reflect the number of infected systems in the network. The assessment of this factor involves the addition of the targets that have already been affected (as reported by Collectors and aggregated by the Detection Engine), and is passed on to the Responder, as part of the Alert.

NAS will be used to assess the Overall Threat to the system after the occurrence of the attack.

6.4.1.8 User Account

The User Account contains information about the account that is potentially associated with the incident, and its associated privileges. If a privileged account were targeted or used as a means to launch an attack, then that attack would be executed with advanced privileges, and thus would have greater potential to cause damage. Information about the User Account, whenever available, should be provided to the Responder by the Detection Engine, as part of the Alert.

The use of a privileged account will raise the Overall Threat of the incident, and consequently justify the use of more severe responses.

6.4.1.9 Suspected Perpetrators

The list of Suspected Perpetrators is assessed dynamically by the Detection Engine, and, whenever available, it is provided to the Responder, as part of the Alert. The related characteristics of each suspected Perpetrator are:

- **Source Address**: the IP Address from which the offending activity originates. An important aspect in the process of locating perpetrators is the use of tracing techniques. Although there are many challenges for their implementation, including high processing requirements, and availability of tracing mechanisms over different (and co-operating) administrative domains, their outcome can provide very useful feedback for the response process. If the Source Address has been used for previous attacks, then the Responder should increase the Overall Threat posed by the incident, and indirectly the severity of the selected Responses. Also, knowing the source of the attack will enable the Responder to issue more effective responses, as close to the source, as possible, aiming to prevent redirection of offending activities against other targets;

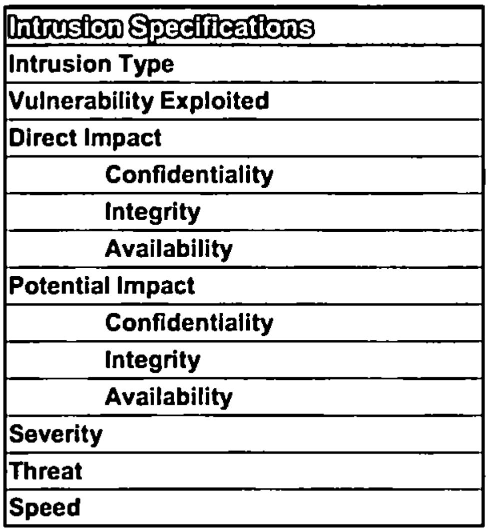
- **User name**: the user name of the offending account (if available). If the perpetrator has used that user account before, the Overall Threat posed by the incident will rise, as the probability of dealing with an attacker will be much higher. Also, by knowing that the user account was created by the attacker, and not a legitimate user, will make the task of issuing severe responses on that account (such as disabling it) easier for the Responder;

- **Insider/Outsider**: the indication of whether the suspected perpetrator is an insider or outsider. If the perpetrator is an insider, the Overall Threat posed by the incident may be higher, as the attacker already has access to the system, and will be able to misuse his rights. Also, if the user is indeed an insider, the Responder will be able to use his User Profile to authenticate him, select most suitable responses that have proved effective before, or restrict his access without disturbing his designated role (e.g.

temporarily restrict web access, but allow him to use Word processing, and spreadsheet software, if his role demands it);

- **Confidence**: a parameter that indicates the probability of having identified the right perpetrator. As some people follow more predictable patterns of behaviour than others, some attackers can be identified more easily than others. Identifying attackers, based upon their behaviour is a new concept, not adequately researched at the time of writing. However, this area has the potential for further research. After all, attackers are humans, and as such, they can be authenticated according to their behaviour, as every other human (Singh 2004; Stoll 1991). The most important obstacle is the lack of enough data to categorise attackers, as they are not regular users of the system. However, deception techniques, and especially honeypots, could prove useful for that task (Honeynet Project. 2003), as their role is to occupy attackers in virtual systems for as long as possible, distracting them from the real targets. The task of identifying attackers would become more achievable if data related to attackers were to be made available within the IT community.

6.4.2 Intrusion Specifications



Intrusion Specifications
Intrusion Type
Vulnerability Exploited
Direct Impact
Confidentiality
Integrity
Availability
Potential Impact
Confidentiality
Integrity
Availability
Severity
Threat
Speed

Table 6.2 Factors retrieved from the Intrusion Specifications

Intrusion Specifications contain information about specific types of intrusions and their characteristics. Once the Detection Engine has indicated the name of intrusion that it believes to have occurred, additional information can be retrieved from the specifications to obtain a comprehensive view of the incident (all of which would again influence the response selection). The type of information retrieved is listed in Table 6.2 and is described in the following sections in more detail.

It should be noted that all the intrusion characteristics are stored in the Intrusion Specifications. Their definition is static, so no refinement needs to take place. However there should be regular updates of the intrusion specifications, in order to include more recent incidents, in the same way that anti-virus and IDS signatures need to be updated and maintained (Symantec 2004).

6.4.2.1 Intrusion Type and Vulnerability Exploited

The Intrusion Type should include the general category, under which the intrusion belongs. For example, the Intrusion Type could be a Virus, Authentication Failure, Buffer Overflow, Denial of Service, etc (see Table 5.1). The Vulnerability Exploited defines which system vulnerability is exploited in the attack. By identifying the relevant vulnerability, the Responder will be able to estimate if the system is vulnerable in the first place, and adjust the Overall Threat level accordingly. Also it can estimate how many other systems could be vulnerable to the same attack, and thus contribute to the estimation of the Number of systems at risk. Finally, it can be used to select suitable responses to patch any systems that are found to be vulnerable.

6.4.2.2 Direct Impact and Potential Impact

The Direct Impact reflects the level of impact that the specific intrusion has upon the confidentiality, integrity and availability of an average system. The Potential Impact reflects the level of impact that can arise if an intrusion is not contained and manages to progress or escalate. For the representation of the Impact, a 10-point scale is used for each of the Confidentiality, Integrity and Availability ratings for systems and data. This approach to rating is common to that used for conventional impact assessment within some risk analysis methods (Flinders University 2004), and the FAIR system uses the 10-point scale to reflect the relative ratings for low (1-3), medium (4-6) and high (7-10) levels, which were already used in the Response-Oriented Taxonomy (chapter 5).

Also, it should be noted that the Direct and Potential Impacts, as specified in the Intrusion Specifications, do not vary for different targets, as they do in the Response Taxonomy. Although it has already been established (from chapter 5) that the same incident can have different impacts upon different targets, the Responder will take that aspect into account at the assessment of the Overall Threat of the incident. In order to do that, the Responder will use the different impact ratings, and the target characteristics, to adjust the Overall Threat. Finally, the impact ratings are used for the calculation of the Severity and Threat, as will be described below.

6.4.2.3 Severity and Threat

The Severity provides a general indication of the direct impacts associated with the intrusion, and can be described as the average of those impacts. The Threat will similarly be represented after considering the individual ratings of the Potential Impacts introduced by the incident. The calculation of the Severity (S) and Threat (T), based upon the Direct and Potential Impacts (D_c, D_i, D_a, P_c, P_i, P_a) is depicted in Equations (5) and (6).

$$S = \frac{D_c + D_i + D_a}{3} \qquad (5)$$

$$T = \frac{P_c + P_i + P_a}{3} \qquad (6)$$

The Severity and Threat are used for the calculation of the Overall Threat, posed by the incident.

6.4.2.4 *Speed*

The Speed reflects the timeframe within which the attack is likely to escalate. That includes the period within which the attack is likely to be repeated against other targets (e.g. in the case of malware), or lead to the occurrence of more incidents, which will probably be more severe. The Speed is represented on a 10-point scale, where intrusions likely to evolve in a matter of minutes are given high (7-10) rating, the ones likely to escalate within hours are given medium rating (4-6), and the ones likely to escalate within days are given lower (1-3) ratings. Finally, the Speed of the attack will be a main influencing factor for determining the Urgency to Respond.

6.4.3 Target Profiles

The Target Profiles contain information about the characteristics of systems within the organisation. After the Responder retrieves the Address of the Target from the Detection Engine, it uses its Profile to retrieve additional characteristics that are relevant for response. The type of information retrieved is listed in Table 6.3 and is described in the following sections in more detail.

Target Profile
Role
Importance
Dependant Systems
Operating Systems
Critical Information?
Critical Files
Critical Operation?
Critical Applications
Other Applications
Auditing Software
Response Software
Usage

Table 6.3 Response Factors retrieved from the Target Profile

The target characteristics are stored in the Target Profiles, and thus their definition is static. However, regular updates of these characteristics would need to take place, in order to account for changes in the systems' environment. The task of updating the Target Profiles could prove to be too cumbersome to be performed manually by an administrator, especially in cases of large organisations. However, it is possible to largely automate the task, either by actively querying systems to determine their characteristics (which could be done at times that do not compromise the systems' and networks' performance), or by inferring the systems' characteristics via passively monitoring their activity. The technology for the first option is already available, and the Responder Agents, installed at hosts, could be enabled to perform that task. Alternatively, a network-scanning tool, such as Nmap, could be used to provide information about the operating systems and running services of hosts in a network (insecure.org 2004). The technology for the passive option, which has the advantage of not introducing any performance degradation on hosts, is not yet as advanced, but represents a new area within the security research domain (Doyle 2003), and has potential for further improvement. In any case, the update of information about some of the factors would still need to be performed manually by the administrator. For example, the role of a system, or its importance, could not be determined by an automated tool. Even in this case though, generic system profiles could be used, to characterise groups of systems with the same characteristics, making the task of updating the individual profiles easier.

The importance of using target profiles lies in the fact that some target characteristics do not change so often, and thus it would be simpler for the Responder to retrieve them locally from a database, rather than having to determine them dynamically, by enquiring the target itself. Also, if the target were unavailable, as a result of the attack, the Responder would not be able to retrieve any information about its characteristics at all.

6.4.3.1 Role

This factor contains information about the role of the target in the organisation. For example, a system might be an 'External Server', 'Internal Server', 'Network Component', or 'User Workstation'. Based upon this information, the Responder can infer the security requirements of that system, in terms of its confidentiality, integrity, or availability. Such information will be particularly important for the adjustment of the Overall Threat, as it will allow the reflection of the different effects incidents can have upon different types of targets.

6.4.3.2 Importance

This factor indicates how important the role of the target is within the organization. For example, the internal database server, which contains all the history of customer transactions in the organisation, is more important than a local print server, since the cost of losing the database server would be much higher. Determining the importance of a system aims to estimate the effect its loss would have upon the normal operation of the organisation, and information of this nature could come from a risk analysis process (Davey 1991). The Importance of a target is represented on a 10-point scale.

The Importance factor gives a general indication to the Responder of how to prioritise its actions, and how to select appropriate responses, according to the system attacked. The more important a target is for the organisation, the higher priority its defence could take. Also, the selection of responses should take into account the target importance. For example, the level of monitoring for important hosts should be higher, and proactive responses, such as actions that aim to prevent escalation of attacks, or safeguard the system assets, should be highly used for such systems. Overall, we can say that the more important a target is in an organization, the more significant the Overall Threat is to the system (in case it is attacked) and the more effective responses are required in a more timely manner.

6.4.3.3 Dependant Systems

The number of systems that depend upon the target is reflected on this factor. It is more relevant for cases of servers, where the number of clients can be estimated. For example, the number of dependant systems of a local file server is the number of hosts in the local area network, whereas the number of dependant systems of the primary DNS Server is more or less all the systems in the organisation. Although a more accurate representation of the dependencies of systems would involve a list of those dependencies, in the same way that they are represented by Toth and Kruegel (2002) in their system (Automated Intrusion Response Model, section 4.4.5), managing such a list would be a very involved process. Although that approach would give a more accurate picture of the environment, as highlighted by Toth and Kruegel, it can only be manageable for a small number of hosts – usually the most important hosts in the network. Thus, a simpler approach was adopted for the FAIR system, and instead of having a list of dependencies within the network, the number of Dependant Systems is used to provide an estimation of the number of systems that could be at risk, if that target were to be compromised, or be made unavailable. That number can be used to determine the Number of Systems at Risk, and consequently the Overall Threat posed by the incident.

6.4.3.4 Operating Systems

This factor contains a list of the operating systems installed on the target, their versions, and most recent updates. By knowing the operating system (and applications) installed, the Responder is able to determine if the system is vulnerable to the attack or not (Internet Security Systems 2003c). Also, it can prove useful for the estimation of the threat posed by the attacker, especially if that attacker has strong history of attacks involving the specific operating system (OS), or is a skilled user of it.

6.4.3.5 Critical Information and Critical Files

The first factor provides an indication of whether the system contains any critical information that needs to be protected. The second factor specifies the location of those files within the system. Files contained in that list will be treated specially, as they might contain confidential information, or information that should not be lost, or modified. Being aware of those files enables the Responder to minimise the effect of attacks and even responses at targets, by making sure that they are protected. The Response Policy should define what should happen to those files in different circumstances, and the options include taking a back up, or restricting access to them to prevent them from being modified or read. Also, the indication of having critical files in a system will trigger the Responder to increase the Overall Threat posed by the attack.

6.4.3.6 Critical Operation and Critical Applications

The first factor indicates whether the system offers any critical operations, and the second factor contains a list identifying them. If the system contains critical operations, then the Responder will adjust the Overall Threat, to account for attacks affecting the availability and integrity of systems. Also, the Response Policy can be configured to restrict access to critical operations (at different levels), update their software to ensure they are not vulnerable, increase monitoring, ensure their load does not exceed a specific threshold, take back up of their status, etc. Overall, the critical information and critical operation factors give a chance to the Responder to identify the most important data and services within systems, and aim to protect them.

6.4.3.7 Other Applications

The list of Other Applications aims to identify applications that represent a high risk, in terms of how easily they can be compromised. For example, according to the SANS 2003 Top 20 List, IIS Server is the software with the most common vulnerabilities (SANS Institute 2003), and thus if such an application were present at a targeted system, that fact should be taken into account. Given that the attacker could take advantage of the presence of such software, the Responder should ensure that it is properly updated, that it has not been compromised already, or that the attacker has restricted (controlled) access to it. Also, the presence of highly vulnerable software will increase the Overall Threat posed by an attack.

6.4.3.8 Auditing Software

The presence of auditing software could provide additional risk for an attacked system, as the attacker could use its output to obtain (or modify) unauthorised information. Thus, the Responder should ensure that access to auditing software by the suspected attacker is either denied completely, or restricted. Also, the Responder could use the outcome from auditing software for its own benefit, to extract relevant information about activity in the targeted system.

6.4.3.9 Response Software

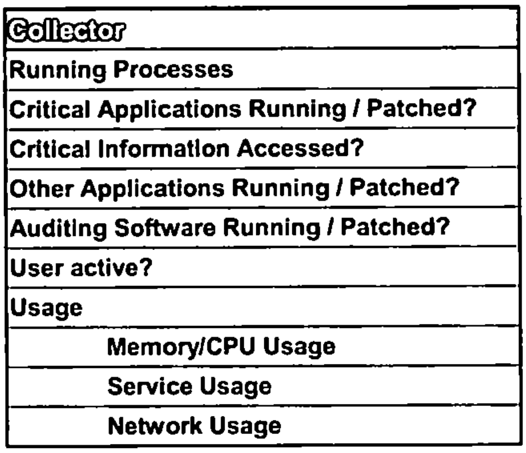
The Response software contains a list of the response modules available on the system. Given the different roles of systems, and the different security requirements for each of them, the response features installed could also vary. For example, some systems might have a web camera, and be able to perform facial recognition, whereas others might have a fingerprint recognition device and software. The Responder needs to know which software is available at the target, in order to select appropriate responses more efficiently.

6.4.3.10 Usage

Knowing the expected usage of systems at given times is not only an important part of detecting abnormalities within them, but also of selecting appropriate responses. The Responder will alter its decisions to suit the usage patterns of a system, whenever possible, in order to avoid disturbing legitimate users and services, and have as little effect upon systems as possible. For example, if a probing event was detected, with a low Speed characteristic, the Responder could elect to check whether the system is vulnerable, and patch any vulnerability, at a time when the overall usage of the system is expected to be low.

6.4.4 Collector

The *Collector* is able to provide information about current activity on the target. This information can be used to minimise the disruption of legitimate activity, by making sure that no important work at the target gets lost, or important applications are not terminated unnecessarily, as a result of selected response actions. It can also be used for cases of compromised targets, when information about them in the Target Profile could be inaccurate, and needs to be reassessed. For example, the determination of whether unauthorised software (e.g. sniffing software, or malware) has been installed will need to be reassessed if the target was suspected to have been compromised. Overall, the Collector helps to minimise the negative impacts of responses, and enhance the response capability as much as possible. The type of information assessed by the Collector is depicted in Table 6.4, and described in the following sections.



Collector
Running Processes
Critical Applications Running / Patched?
Critical Information Accessed?
Other Applications Running / Patched?
Auditing Software Running / Patched?
User active?
Usage
Memory/CPU Usage
Service Usage
Network Usage

Table 6.4 Response Factors assessed by the Collector

6.4.4.1 Running Processes

The Collector retrieves the list of running processes in the target, mainly to detect the presence of any unauthorised software. Whenever there is a suspected case of a target being compromised, then it is possible for unauthorised software to be present, and the Responder should take account of that. If such software were detected, then more severe responses would be warranted, depending upon the characteristics of the software. For example, according to the Response Policy, the Responder could decide to disconnect the system from the network, mirror an image of its hard drive and memory for forensic purposes, run integrity checks, take appropriate back ups of critical information and services, and restore the system to its original state, as much as possible.

Also, the Collector is able to detect whether any of the running processes are critical applications, highly vulnerable software, auditing software, or are used to open critical files.

6.4.4.2 Critical Applications Running

As the Collector retrieves the list of running processes on the target, it is able to detect if any Critical Applications are running at the time. Knowing whether critical applications are running could trigger the Responder to take a back up, increase monitoring, restrict user access to only the necessary functions (not allowing login of a privileged account, any configuration changes, access to log files, access to configuration files, user login etc), or block access to the application altogether and terminate it, if it is not busy at the time (and its expected usage is low as well). Also, the Collector can enquire if the latest patch has been applied on the application, to make sure that it is not vulnerable. The reason for doing that would be to ensure that normal updating procedures have indeed been performed, and not been cancelled, postponed, disabled by a user, or been aborted by the updating module due to network problems.

6.4.4.3 Critical Information Accessed

If the application used to open a critical file is running at the target, then the Collector will enquire whether that file is already open by that application. If it is, then depending upon the context of the attack in progress, the Responder can select to deny user access to the critical file, take a back up, or save a copy of the open file before issuing a severe response (such as disconnecting the user, or shutting down the application). Also, the Responder could increase monitoring of activity, to ensure that the file is not modified, or deleted.

6.4.4.4 Other Applications and Auditing Software Running

If Other Applications are running, then the Responder could chose to check whether their version is fully updated and patched, and depending upon the role of the application (whether it is critical or not), it can be terminated, or patched as soon as possible.

Similarly, the presence of auditing software in the running processes could trigger the Responder to check whether it is fully updated and patched, and depending upon whether it is a critical application, to terminate it, or restrict access to it by the suspected process / user. Alternatively, additional monitoring could be elected, to ensure that the suspicious activity does not evolve to cause additional attacks.

6.4.4.5 User Active and Usage

The Collector is able to determine if the user connected to the target is active or not. There is an active user in the system if mouse or keystroke activity occurs within a specified timeframe. Inactive status is determined as long as there is not any keystroke or mouse activity in the system for more than the specified timeframe. The Responder can use that information to minimise the impact of response actions, by initiating them at times when no user will be interrupted. For example, in the event of the Responder deciding that update of software is needed to patch existing vulnerabilities, the task can be performed either immediately (risking disruption of legitimate users and services), at a later time (when the expected load of the target is low), or as soon as the user becomes inactive. Scheduling the update to happen when the expected load is low might be too late, depending upon the speed of the attack, whereas the latter option has the potential to be more effective.

Apart from user activity, the Collector is able to monitor the overall system activity, mainly reflected in the load of network usage, service usage (dependant upon statistics of specific applications), memory usage and CPU usage. The network usage monitors the load of incoming and outgoing packets, giving an overview of how busy the target is at the time. That metric is especially relevant for servers and network components, for which network bandwidth consumption is a clear indication of how busy they are at the time. The memory usage is the percentage of used memory in the system, and the CPU usage is the average of the CPU usage over a short period of time. The calculation of the memory usage and CPU usage is depicted in equations (7) and (8). The M_{use}, M_{free}, M_{avail}, M_{Total}, CPU_{use}, CPU_{usage}, and n represent the percent of Memory usage, percent of Free Memory, volume of available memory in Mbytes (Physical and Pagefile), volume of Total Memory in Mbytes (physical and Pagefile), average of CPU Usage, individual rating of CPU usage, and number of individual ratings, respectively.

$$M_{use} = 100 - M_{free} \Rightarrow M_{use} = 100 - \left(\frac{M_{avail}}{M_{Total}} * 100 \right) \qquad (7)$$

$$CPU_{use} = \frac{\sum_{i=0}^{n} CPU_{usage}}{n+1} \qquad (8)$$

Obtaining this sort of information can enable the Responder to determine if server applications are busy in the system, and whether (or when) they can be interrupted. For example, high CPU usage denotes a busy system, whereas high memory usage potentially denotes a system with many open applications. The Responder could increase its confidence to issue more severe responses (such as disconnecting the user), issue responses that require system restart, or shut down the system, at times when the memory and CPU usage are low. In such a case, even if all the applications' status were to be saved before their termination, there would still be a level of disturbance by terminating them. However, fewer applications would be disturbed, and so the impact of the response would not be so significant.

6.4.5 Attacker Profiles

The consideration of historical profiles of attackers involves the build up of information about attackers who have previously targeted the system, the methods they used, and the responses selected to counteract them. Information about known attackers can be particularly useful in the intrusion response process, since:

- suspicious attacks, which appear to be carried out by known attackers can be treated with higher suspicion, as there are more chances of such a person attempting unauthorised activity again. Actually, even the presence of a known attacker in the network (who might not have done anything suspicious yet) should raise the level of suspicion; it is quite likely that he has already compromised a target (without being detected) or is about to do so in the future. Thus the responder could increase the monitoring level, check for vulnerability updates, or check the integrity of involved systems.

- the level of damage caused by attackers, can indicate the degree of threat the system is under; the more serious that threat, the higher the Overall Threat will be. The danger of the attacker is represented on a 10-point scale, and one way of assessing its value would be to consider the distribution of attack costs, so that an attacker causing average damages (according to the DTI survey from section 2.2, the average damage was reported as £30,000) would have a rating of 5, and so on.

- the competence level of an attacker can also be used to influence the Overall Threat posed by the incident. The assessment of the competence level is more likely to be performed by a human, as part of the forensic analysis of verified attacks. The assessment can be based upon the methodology and knowledge of the attacker about the targeted systems. Skilled attackers do not waste time looking for unnecessary information, do not make extra steps by experimenting on different attack strategies, nor generate many errors (Honeynet Project 2000). They often utilise sophisticated characteristics of systems, specific to certain versions of operating systems and software and, more often, do not leave any tracks of their activity (Phreak Accident 1993).

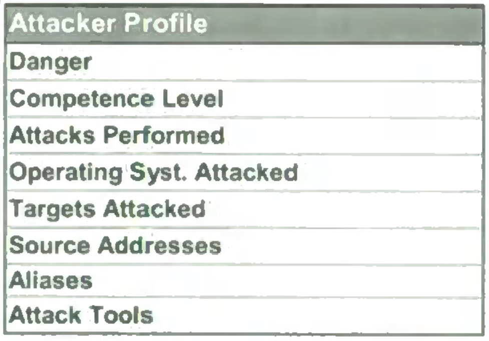
- it is possible to protect systems by making sure that they are not vulnerable to the attacks the specific perpetrators have attempted before. Surely, if the person in question has used specific tools to attack targets before, then it is likely that he will use them again, given the chance. Thus the system could ensure that all relevant vulnerabilities have been successfully patched and that the tools possessed by the attackers cannot introduce any additional threat to the system. Identifying the activity of attack tools can sometimes be automatically detected by the IDS, or as part of the forensic analysis, where the target is analysed for the presence of such tools.

- correlating information such as the Source Addresses and Targets the attacker has previously utilised and / or attacked, with the outcome of tracing mechanisms, will increase the effectiveness of responses. Identifying potentially compromised or offending systems will enable the Responder to respond appropriately and correct problems as close to their source as possible.

- it is possible to provide forensic evidence that can link specific attackers with attacks and thus enable the organisation to follow legal prosecution, if such a need arises. Even if the Responder elects not to act upon the presence of a known attacker in the network, it could at least increase the monitoring level at the relevant hosts, and collect evidence about the attacker's activity. One aspect of that is the extraction of usernames and passwords the attacker chooses when selecting names for accounts. This information could be used to identify attackers, and correlate incidents with similar characteristics. The idea behind this is to identify cases of attackers using the same 'alias' (or nickname) in usernames, or passwords, when they create new accounts in a system, or when they log in to chat networks.

Having said all this, one can argue that of all the contextual factors, using attacker profiles is the hardest to achieve in practice. Firstly, as already discussed, it is very difficult to retrieve these characteristics in the first place. Also, the information about them is often not accurate, as they are likely to use spoofed IP addresses, masquerade by using stolen / hijacked system accounts and use a chain of connections to hide their tracks. Still though, if the Responder is confident that the suspected activity is indeed malicious, then instead of disconnecting or blocking the user / process, it could redirect the session to a decoy system, which will replace the target. In the decoy system, methods of automatically tracing attackers, or different authentication profiling methods to successfully identify them can be performed.

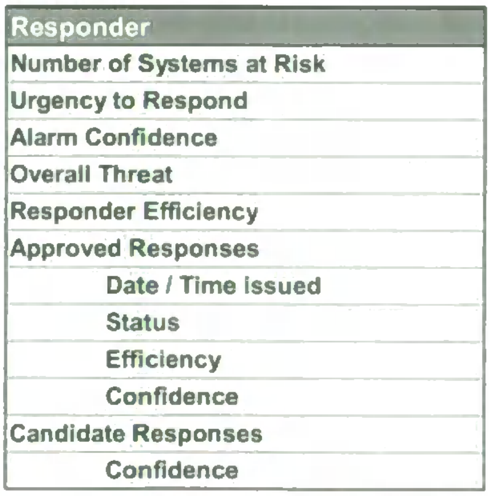
As mentioned before, one can argue that there is not much value in keeping information about attackers who have previously targeted systems in the organisation, because there is no information (or related research) indicating that attackers tend to prefer attacking the same systems they have attacked before. However, there are still chances of them attempting to attack other systems within the organisation, or other partner-networks. Also, if information about attackers were shared within a greater number of organisations (for example, share information via a third-party reporting organisation), then the value of such information could become even more significant. The type of information that could be collected about attackers is summarised in Table 6.5.



Attacker Profile
Danger
Competence Level
Attacks Performed
Operating Syst. Attacked
Targets Attacked
Source Addresses
Aliases
Attack Tools

Table 6.5 Response Factors contained in the Attacker Profile

6.4.6 Responder



Responder
Number of Systems at Risk
Urgency to Respond
Alarm Confidence
Overall Threat
Responder Efficiency
Approved Responses
Date / Time issued
Status
Efficiency
Confidence
Candidate Responses
Confidence

Table 6.6 Response Factors assessed within the Responder

After receiving contextual information from the Detection Engine, and the Collector, the Responder assesses the following factors, depicted in Table 6.6.

6.4.6.1 Number of Systems at Risk

The Number of Systems at Risk is assessed dynamically by the Responder, and it contains the number of systems that are in danger of facing a security threat after the occurrence of the specific attack. Their assessment involves the estimation of the number of systems depending upon the target (that is how many systems would be disrupted, if the target became unavailable), and the number of systems that could have the same vulnerability as the one affected (i.e. implying they could also be targeted in the near future). The latter estimation involves retrieving, from the System Profiles, the number of systems using the operating systems and / or applications affected by the vulnerability.

This factor will be used to reflect the extent of the incident, and assess the Overall Threat of the attack. The Number of systems at risk can also be used for the determination of appropriate responses that will prevent such further systems from being targeted. Specifically, according to the number of systems that might need to be updated, and the urgency of needing to do so, the Responder can elect to update them immediately, or as soon as these systems become inactive, or later on (for example after 11:00 PM), when the systems are expected to have low load.

6.4.6.2 Urgency to Respond

This factor relates to the urgency with which a response action is needed. It is assessed dynamically by the Responder, and it partly reflects the speed of the attack. It is also weighed by considering the dynamics of the incident, which relates to the volume of activity associated with it. The hypothesis behind it is that, if an incident is already associated with a large volume of activity, then it is likely that similar volumes of activity will follow in the near future, possibly leading to escalation, or progression of attacks. Thus, the need to respond in a timely manner would be greater. In order to estimate the dynamics of the incident, the number of alerts already generated for the same incident is calculated. The actual Urgency to Respond is calculated by increasing the Speed of attack, either by 20%, 40%, or 60%, according to the number of alerts. In order to do that, a user-defined threshold is used to define three bands, against which the number of incidents will be compared. The threshold for each band is determined by multiplying the user-defined threshold by 1, 2, and 3 respectively. If the number of alerts fits within the higher band (exceeds the user-defined threshold multiplied by 3), then the Speed of attack is increased by 60%. If it fits within the medium band (exceeds the user-defined threshold multiplied by 2), then the Speed of the attack is increased by 40%, and if it exceeds the user-defined threshold itself, it is increased by 20%. For example, if the administrator sets the threshold to be 10, and the Speed of the attack is 5, then 11 to 20 alerts within an incident would increase the Speed by 20%, and thus make the Urgency 60. If there were 21-30 alerts, the Urgency would become 70, and for over 30 alerts within the incident, the Urgency would become 80. Equation (9) depicts the calculation of the Urgency, where N_{Inc} and Thr represent the number of alerts in the incident, and the user-defined threshold respectively. It should be noted that Equation (9) is not derived from experimental analysis of incident case studies, and thus the factors used, and indeed the formula itself could be different. However, it can serve its purpose as a starting point, providing an indication of how the Urgency can change according to the speed of the attack, and the volume of malicious activity introduced by it.

$$Urgency = 10 * Speed * (1 + (2 * factor)) \qquad (9)$$

Where factor is derived as follows:

$$N_{Inc} > (Thr * 3) \Rightarrow factor = 0.3$$
$$(Thr * 2) < N_{Inc} \leq (Thr * 3) \Rightarrow factor = 0.2$$
$$(Thr) < N_{Inc} \leq (Thr * 2) \Rightarrow factor = 0.1$$

The Urgency is represented on a 100-point scale, as all the factors assessed by the Responder. Since most of these factors receive multiple inputs from other factors, they are likely to vary more, and thus it is more suitable to represent them in a higher scale.

According to the urgency to respond, a different course of response actions will be initiated by the Responder.

6.4.6.3 Alarm Confidence

The Alarm Confidence reflects the confidence that the alarm is indeed a true attack, after considering the Intrusion Confidence and the Detection Efficiency. The Alarm Confidence is depicted in Equation (10), where C_{Alarm}, $C_{Intrusion}$, and DE represent the Alarm Confidence, Intrusion Confidence, and Detection Efficiency respectively.

$$C_{Alarm} = \frac{C_{Intrusion} * DE}{100} \qquad (10)$$

Alarms originating from less capable detection engines (or engines creating a high number of False negative alarms) will need to obtain considerably higher level of confidence, in order to warrant the issue of severe response actions. The Alarm Confidence will principally be used to adjust the Overall Threat posed by the incident. That process will also involve the consideration of several other factors, and will be discussed in the next section.

6.4.6.4 Overall Threat

The Overall Threat represents the danger that arises after the occurrence of an incident, and is represented on a 100-point scale, as it receives input from a large number of contextual factors. There are three main aspects that are considered for the calculation of the Overall Threat. These are the threat arising from the Incident, the threat arising from the target (the insecure characteristics of the target that could present potential problems), and the threat arising from the perpetrator.

The steps towards determining the Overall Threat are described below:

- **Intrusion Severity, Intrusion Threat** – The weighting of these factors was increased, since the large number of factors used to calculate the Overall Threat tended to overshadow the importance of the Intrusion Severity and Threat.

- **Role of Target, Intrusion Impacts** – If the target is an internal server, or network component, then the threat is increased, according to the confidentiality, integrity and availability impacts of the incident. If the target is an external server, then the threat is increased, according to the availability and integrity impacts of the intrusion.

- **User account** – If the user account is privileged, then the threat is increased, according to the direct and potential impacts of the intrusion.

- **Target has Critical Information** – If the target has critical information, then the threat is increased, according to the confidentiality, integrity, and availability impacts of the intrusion.

- **Target offers Critical Operation** – If the target offers a critical operation, the threat increases, according to the integrity and availability impacts of the intrusion.

- **Alarm Confidence** – The threat is adjusted according to the alarm confidence. That involves multiplying the current sum, which reflects the threat posed by the intrusion, with the alarm confidence by 100 (Threat = Threat*Alarm Confidence / 100).

- **Target has Auditing – Other Software** – If the target has auditing or highly vulnerable software, the threat increases according to the importance of the target.

- **Number of affected systems - Number of systems at risk** – According to how they compare to the total number of systems in the network, the threat levels are adjusted accordingly.

- **Perpetrator Threat** – The Threat level for each perceived perpetrator is assessed, by considering his danger level, competence level, the volume of attacks he has performed, and so on. The threat for each perpetrator is adjusted to account for the suspected perpetrator's confidence.

- **Target vulnerable to attack** – If the target is not affected by the attack, the Overall Threat is adjusted so that it does not exceed 30 (which is the threshold for low Overall Threat values). Hence, it is multiplied by 0.3.

The determination of the Overall Threat will be mainly used for the assessment of the Alert Status and for the selection of appropriate response actions.

6.4.6.5 Responder Efficiency

This metric reflects the capability of the Responder to handle intrusion alerts and make the right decisions. As mentioned in section 6.3.1, the higher that capability is, the more autonomy the Responder can have, especially in cases when severe responses are necessary. Overall, the Responder Efficiency is a generic metric that applies to the Responder and aims to improve its operation, and the level of effect it can have upon systems, users and networks. The Responder Efficiency is represented on a 100-point scale, and is refined gradually to reflect the changing performance of the Responder. It is calculated, according to the equation (11), where R_{Eff}, n_{pos}, and n_{Total} represent the Responder Efficiency, number of positive (correct) decisions, and total number of decisions respectively.

$$R_{Eff} = \frac{n_{pos}}{n_{Total}} * 100 \qquad (11)$$

6.4.6.6 Approved Responses

After the contextual factors are assessed, and the Responder makes a decision, the outcome of that decision will be two sets of responses. The first set is the Approved Responses, which contains the list of responses authorised to be issued by the Responder. The time these responses are issued is stored for reference, and for monitoring their effect according to the timeframe in which they are issued. Also, their status is stored, in order to monitor their outcome. As soon as the Responder requests them to be issued, the responses are stated as 'Issued'. Their status can then change to 'Completed', 'Pending', 'Aborted', as soon as they are completed, delayed, or aborted for some reason.

The administrator can also inform the system about the efficiency of the issued responses, providing comments at the same time, about their performance. These comments provide feedback about whether the responses were successful to fulfil their purpose, too severe, not severe enough, whether they had unwanted side effects, whether they were applied too late, whether they were completely inappropriate, whether they were applied in a false alarm scenario, or whether they were not completed for some reason. As discussed in section 6.3.2, knowledge about the appropriateness of response actions will enable the Responder to adjust the efficiency levels of the response actions, and the Responder itself. Also, it will provide the basis for improving these levels, either with the help of the administrator, or machine-learning algorithms.

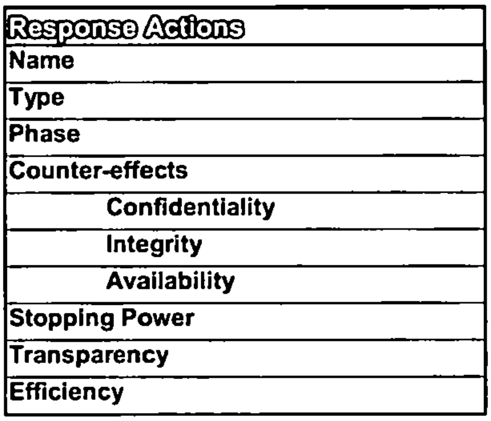
A final metric that is stored for each issued response is the confidence level with which it was recommended. This level is derived by considering which characteristics would constitute an 'ideal' response, and grading how compatible the issued response is to those characteristics.

6.4.6.7 Candidate Responses

The second set of responses is the Candidate Responses. They contain the list of actions, with lower confidence, either because their characteristics are not such a strong match to the 'ideal' responses, or because the Responder Efficiency is not high enough for them to be authorised. Responses contained in this list need prior human authorisation in order to be issued. As soon as the administrator authorises them, then they are added to the 'Approved Responses' list. Since these responses are not actually issued yet, the only characteristic that is stored about them is the confidence level with which they are recommended.

6.4.7 Responsc Actions

Information stored in this module relates to the characteristics of Response Actions available by the FAIR System. These are retrieved by the Responder, in order to select the responses with the most appropriate characteristics. Unless otherwise stated, the definition of response characteristics is static (no refinement needs to take place). However there should be regular updates, in order to include new response actions. These characteristics are listed in Table 6.7, and are described below.



Response Actions
Name
Type
Phase
Counter-effects
Confidentiality
Integrity
Availability
Stopping Power
Transparency
Efficiency

Table 6.7 Response Factors stored in Response Actions

6.4.7.1 Name

This factor contains the name of the response action. The administrator is able to use the response names to customise the Response Policy, and determine which responses would be suitable for which attacks, under which circumstances. The customised policy can be as generic, or as specific, as the administrator wants. The Responder will use these indicated responses as a starting point, and consider their characteristics to select the ones that are more suitable. These characteristics are described below.

6.4.7.2 Type and Phase

The Response Type reflects whether the specific response is passive or active (see sections 3.1.1 and 3.1.2). The Phase reflects the main objective of a response. Relating back to the aims of response, summarised in section 3.1 (notifying the administrator, collecting more information about the incident, protecting system resources in the long and short term, and identifying the perpetrator), the phases of a response necessary to achieve these aims are:

- Notify
- Investigate
- Protect Resources
- Recover
- Collect Evidence, and
- Forestall potential problems

In the FAIR System, each response action is associated with one of the above phases. One could argue that a response could be used in more than one phase (so for example, patching vulnerabilities could be used to protect resources, recover, and forestall potential problems), and thus, a list could be used to contain all the phases appropriate for each response. However, for simplicity, the phase value contains the main purpose for using a specific response, so in the case of patching vulnerabilities, its phase would be Forestall.

The Responder, based upon the Response Policy, will decide which phases are more appropriate for the specific alert, by combining information from contextual factors. For example, if the Alarm Confidence and the Urgency to respond were not high, then the best Phases of response would be to Investigate, and Collect Evidence. If the Alarm Confidence and the Responder Efficiency were high, then the best Phases would be Protect, Collect Evidence, and Recover.

6.4.7.3 Counter-Effects

The Counter-Effects partly reflect the side effects of a response action. In other words, they represent the impact they can have upon the Confidentiality, Integrity, and Availability of systems and data. They are represented on a 10-point scale for each of the Confidentiality, Integrity and Availability ratings. Responses that might give in information about the presence of the IDS, software versions, system vulnerabilities, or important assets of a system will have impact upon the confidentiality of that system. Responses that affect the integrity of systems, especially in the case of deceptive responses (where false information is given to suspected attackers), or recovery responses, usually have high integrity counter-effects. Finally, responses that deny access to systems, processes, or data, normally have high availability counter-effects.

As the Responder selects the most appropriate characteristics of candidate responses, the counter-effects should be as low as possible. So, if two response actions with similar characteristics are selected, the one with lower side effects will receive higher level of confidence. Considering the counter-effects of a response is particularly important in case of false alarm scenarios, when in other words the Alarm Confidence is low. Generally, low counter-effects are considered important when the Overall Threat and Urgency are low, as there is no pressing need for the FAIR system to interfere significantly. By contrast, when the Overall Threat and Urgency are high, the Responder will need to select more effective responses, even ones with high counter-effects.

6.4.7.4 Stopping Power

This metric reflects the perceived strength of a response (its likely 'stopping power') against the attack. It is represented on a 10-point scale, where the different ratings are derived according to Table 6.8.

Response Stopping Power	
9-10	Block / Stop Attack
7-8	Redirect
5-6	Stop partially / Limit
4	Postpone, Delay
3	Investigate
2	Collect Evidence
1	Minimise loss at target
0	None

Table 6.8 Response Stopping Power Ratings

In another implementation scenario, where the same response could be used for more than one purposes (e.g. Delay attack, and Investigate), separate ratings for each purpose could be applied. For example, a response could have a rating of 7 for partially stopping the attack, and 5 for delaying it. However, this approach would be very involved, and possibly too complicated for the scope of this research. Thus, in the FAIR System, there is only one rating for the Stopping Power, which reflects the main purpose of the response.

The Responder selects the maximum Stopping Power allowed, and no approved or candidate response will have higher Stopping Power than the one determined. The maximum level of Stopping Power is determined in the Response Policy, and influenced by the Urgency, Overall Threat, Responder Efficiency, Alert Status, and Target Importance. The higher these factors are, the higher the maximum Stopping Power can be.

6.4.7.5 Transparency

The Transparency reflects how apparent a response will be to the attacker, legitimate users, or the system overall. Responses such as collecting evidence have high transparency, as the attacker or normal users are not aware of them being issued. On the contrary, explicit authentication requests, such as using cognitive questions, have low transparency, as the user is very much aware of the presence of the response system, and cannot go on with his activities unless he replies to the questions. The transparency is represented on a 10-point scale, and the aim of the Responder is to select responses with as high transparency as possible. Like the counter-effects, transparency is particularly important when the Urgency, Overall Threat, and consequently the Alarm Confidence are not high. When these factors have high values, then the Response Efficiency is considered more important, and the higher Efficiency a response has, the more suitable it will be. The Response Efficiency is discussed below.

6.4.7.6 Response Efficiency

The Response Efficiency reflects the overall effectiveness of the response action, based upon its historical performance, and specifically the feedback received whenever it is issued (see section 6.4.6.6). It is thus refined gradually, to reflect additional feedback from new alerts.

The Response Efficiency is represented on a 100-point scale, and the higher it is, the more suitable a candidate response can be. In fact, as mentioned earlier, it is particularly important to select highly efficient responses in cases of high Urgency, and high Overall Threat, even if their counter-effects and transparency are not ideal (comparing to other candidate responses).

6.5 Response Policy

As already discussed, the Response Policy is the expert system module, which is responsible for identifying the most appropriate response characteristics, according to the context of the attack. After the occurrence of an alert, the Response Policy receives input from the Responder about the static and dynamic context of the attack, and according to that context, it selects which characteristics would be more suitable for the selected responses. The rules used in the Response Policy have been derived from the author, and aim to determine several considerations/responses that should be indicated at different scenarios. More details about these rules can be found in Appendix C.

As a starting point, the first characteristic identified is the Response Name. The administrator is able to fully customise this part of the selection process, via a user-friendly interface. The interface, with which it is possible to customise the Response Policy, allows the association of responses with attacks, and possible conditions under which they can be selected (the conditions relate to the factors already identified within the chapter). For example, the Response Policy could define rules, such as:

```
"If the Alert relates to a Buffer Overflow attack,
 and the Target is Vulnerable, then
      Deny / Stop the action,
      Patch the Vulnerability immediately"
"If the Alert relates to a Buffer Overflow attack,
 and the Target is Not Vulnerable, then
      Check how many systems are vulnerable,
      Patch the Vulnerable Systems as soon as possible"
```

The presence of conditions is not necessary, leaving the flexibility for the use of as simple or as complicated response policies as possible. For example, for a Suspicious Login alert, the Response Policy could include the following rules.

```
"If the Alert relates to a Suspicious User Login, then
     Issue Keystroke Analysis,
     Issue Continuous Keystroke Analysis,
     Issue Cognitive Questions Authentication,
     Issue Facial Recognition,
     Restrict User Access,
     Disconnect User"
```

Or

```
"If the Alert relates to a Suspicious User Login
 and the User-Account Is Privileged, then
     Issue Continuous Keystroke Analysis,
     Issue Cognitive Questions Authentication,
     Restrict User Access,
     Disconnect User"
"If the Alert relates to a Suspicious User Login
 and the User-Account Is Not Privileged, then
     Issue Keystroke Analysis,
     Issue Facial Recognition,
     Restrict User Access"
```

Based upon the user-defined policy, a number of Response Names are selected as candidates. Then, more general rules are applied to determine the remaining response characteristics. These rules are not specific to the different types of attacks, and cannot be as easily customised by the administrator. Such a task would involve direct modification of the rules file and would thus require knowledge of the rules syntax. The fact they cannot be so easily modified is justified by the fact that they are more generic rules, and thus are not expected to need modifications very often. Overall, there are three sets of rules, which aim to determine:

- the most appropriate Response Phases,
- the maximum level of Stopping Power allowed, and
- how important the Response Efficiency should be, in comparison to the response side effects (Transparency and Counter-Effects).

The rules to select the most suitable Response Phases are mainly influenced by the Responder Efficiency, Overall Threat, Urgency, and Alarm Confidence. The nature of these rules can be illustrated in the following example.

```
"If Responder Efficiency is Low,
 and Overall Threat is High, then
      suitable phase is Notify"
"If Alarm Confidence is Low,
 and Urgency is Low, then
      suitable phases are Investigate and Collect Evidence"
"If Alarm Confidence is High,
 and Responder Efficiency is High, then
      suitable phases are Protect Resources, Collect Evidence, and
      Recover"
```

The first rule suggests that notification alerts are suitable for cases when severe responses are probably needed, and the Responder is not able to issue them automatically (because the Responder Efficiency is low). By notifying the administrator, he will review the decision and authorise any severe responses that the Responder could not issue. The second rule suggests that if there is a suspected attack with low confidence, and the probability of it escalating rapidly is low, then the best phase of response would be to Investigate and Collect Evidence. In that way, more information about the incident will be collected, aiming to determine whether the attack is really occurring or not. Finally, the third rule suggests that if the Alarm Confidence and the Responder Efficiency are both High, then the most suitable phases are to Protect Resources, Collect Evidence, and Recover.

The maximum level of Stopping Power is determined by considering the Responder Efficiency, Alert Status, Urgency, Overall Threat, and Target Importance. Effectively, the higher these factors are, the higher the Stopping Power can be. Since the Responder Efficiency and Alert Status are more generic metrics, not specific to the occurring incident only, they receive higher weighting, so that if both of them are low, no severe responses can be issued, regardless of the value of the other factors. So, if both the Responder Efficiency and Alert Status are low, the Stopping Power cannot exceed 7 (or even 4), even if the Urgency, Overall Threat and Target Importance were High.

Finally, the last part of the selection process is to determine the weighting of the Efficiency and side effects of a response. Influencing factors for that process are the Overall Threat and Urgency. The higher these are, the more important it is for responses to be efficient, and the lower they are, the more important it is for responses to have low side effects.

After the 'ideal' response characteristics are determined, the candidate responses (the responses already selected in the first phase) are given confidence metrics, according to how closely they match the desired response characteristics. The strongest choices (the ones with confidence higher than 50%) are included in the list of approved responses, whereas the rest remain in the category of candidate responses. As already described, the approved responses are the ones to be issued, while the candidate responses are the ones that are included in the alert as a reference for the administrator.

6.6 Conclusions

This chapter has focused upon the conceptual architecture for a Flexible Automated Intelligent Responder system. Its description has included an introduction of the main concepts of the architecture, and the modules within it. Detailed focus was given to the role of each module, and especially its contribution in the response decision process. The central point in the process is that of the contextual factors influencing response, and thus the way they are assessed and used within the system was described. Finally, the Response Policy was presented, aiming to describe how each of the elements can be utilised within the context of the response policy.

Although determining the aspects of the FAIR system has been a long process, it was important to illustrate how novel aspects of the system would be defined and used. Having established this, it is still necessary to evaluate the practical viability of such a system, and demonstrate how the main features of FAIR would operate in a practical scenario. As such, the next chapter presents the implementation of the FAIR prototype system, which aims to demonstrate that aspect of the research; the fact that the FAIR System is viable and can be implemented in practice.

CHAPTER 7

A PROTOTYPE AUTOMATED RESPONDER

7 A PROTOTYPE AUTOMATED RESPONDER

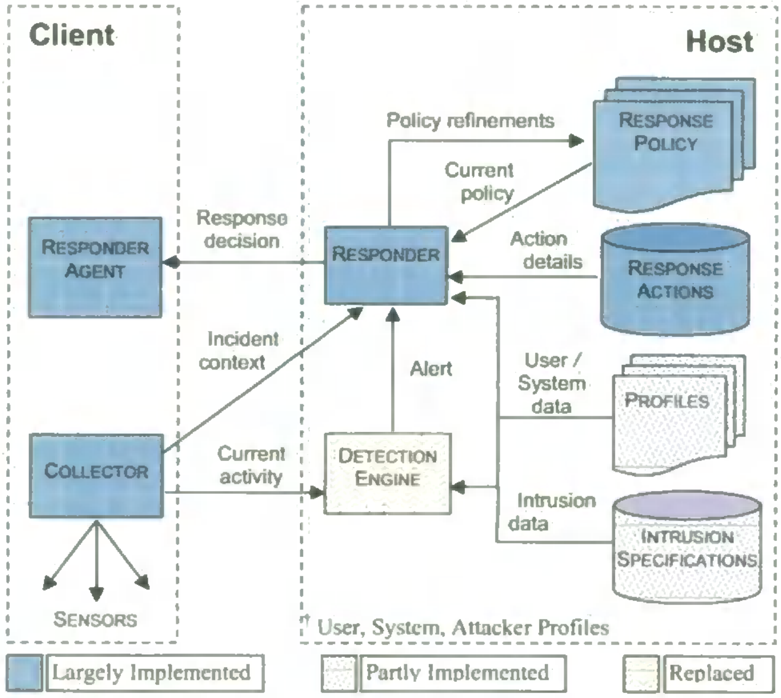
7.1 Introduction

This chapter describes the implementation of a prototype system, which embodies a subset of the key elements of the proposed architecture; namely the ability to adapt decisions according to changes in the environment, and provide easily customisable response policies. The aim of the prototype system is to demonstrate the principal concepts of the proposed architecture. The description in this chapter essentially details the most important features of the prototype, highlighting the aspects of the FAIR architecture that have been realised in practice. Different attack scenarios are then analysed, demonstrating the effect of contextual factors upon the response decision process, and how the Responder can adapt its decisions to reflect changes in the environment (including changes in the Responder and the targeted system). Finally, the interface of the Response Policy Manager is described, highlighting its role in the process of customising Response Policies.

7.2 Implementation Overview

The elements of the FAIR architecture that have been implemented in the prototype are depicted in Figure 7.1. Since aim of the implementation has not been to produce a fully functional system, but a proof-of-concept tool, none of these elements have been fully realised. Instead, focus has been given to the features that assess the context of an attack, adapt decisions of the Responder based upon that context, and customise the Response Policy to reflect experience from previous incidents. Hence, the degree to which each element has been implemented depends upon its role in the achievement of these features. Specifically, the Responder, Response Policy, Response Actions, Responder Agent, and Collector have been largely implemented, whereas the Profiles, and Intrusion

Specifications have been implemented to a much lesser degree. Finally, in the absence of an IMS Detection Engine (the IMS is a conceptual architecture, with not all its modules realised yet in practise), the Detection Engine has been replaced by an attack simulation interface, which was developed as part of this work and is described in section 7.3. It should be noted that a full implementation of the FAIR system would require further research in its own right, in order to extend and refine in more detail the role of several elements, such as the Attacker Profiles.



† User, System, Attacker Profiles

Figure 7.1 Prototype Implementation

The prototype system consists of 3 modules, namely the Alert Simulation Console, the Responder, and the Responder Agent. As indicated above, the Alert Simulation Console aims to replace the role of the Detection Engine. The Responder incorporates the functionality of the Responder, Response Policy, Response Actions, Profiles, and Intrusion Specifications. Finally, the Responder Agent sits at the target of the attack, and encompasses the functionality of the Responder Agent, and Collector. The 3 modules of the prototype system are illustrated in Figure 7.2. In this figure, after the Alert Simulation Console sends an alert to the Responder, the latter communicates with the Responder Agent to either request for more information about the alert, or request specific responses to be issued. In each case, the Responder Agent informs the Responder about the outcome of the request.

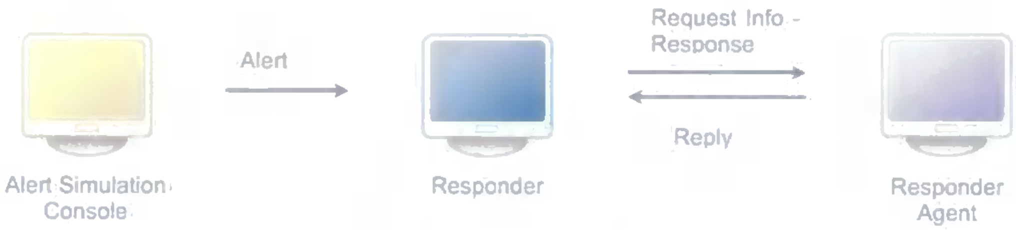
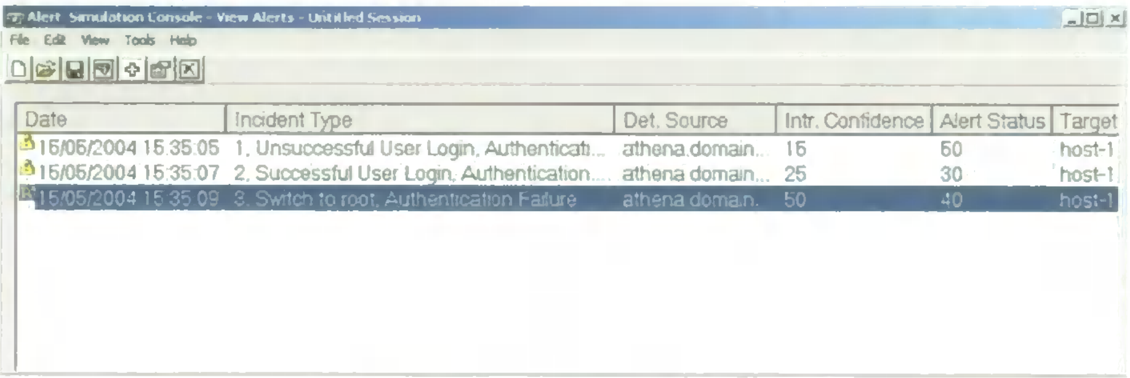


Figure 7.2 Prototype System Modules

For the prototype, the Alert Simulation Console, Responder and Responder Agent have been implemented for Windows XP platforms, which is the operating system supported within the University of Plymouth. Visual Basic 6.0 was used for the software development, due to its user-friendly interface, and the wide documentation available, which shortened the software development phase, without limiting the functionality of the FAIR prototype. The expert system module was based on CLIPS (Giarratano and Riley 1998), and an ActiveX control, which provided the communication between Visual Basic and CLIPS.

7.3 Alert Simulation Console

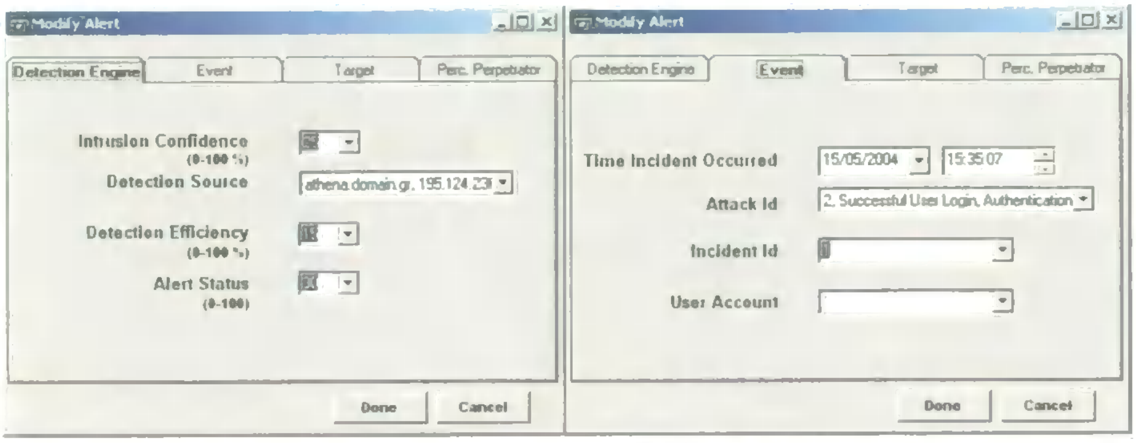
In the absence of a full Detection Engine, or indeed genuine incidents, the Alert Simulation Console is necessary to enable incident alerts to be generated and sent to the Responder, in order to trigger its involvement. The main interface of the console is depicted in Figure 7.3, in which alert simulation sessions can be created, opened, or saved. These sessions contain a selection of alerts with various characteristics, one of which is the time they are meant to have been generated. Once a session is opened, or created, all the alerts in that session are sorted according to that time, and when the user selects to generate the alerts, the console will send each of them to the Responder at time intervals, equal to their respective time differences. In that way, intrusion scenarios can be simulated, enabling the Responder to receive multiple alerts in the same way that it would have received them, had they originated from a genuine Detection Engine.



Date	Incident Type	Det. Source	Intr. Confidence	Alert Status	Target
15/05/2004 15:35:05	1. Unsuccessful User Login, Authenticati...	athena.domain...	15	50	host-1
15/05/2004 15:35:07	2. Successful User Login, Authentication....	athena.domain...	25	30	host-1
15/05/2004 15:35:09	3. Switch to root, Authentication Failure	athena.domain.	50	40	host-1

Figure 7.3 Alert Simulation Console

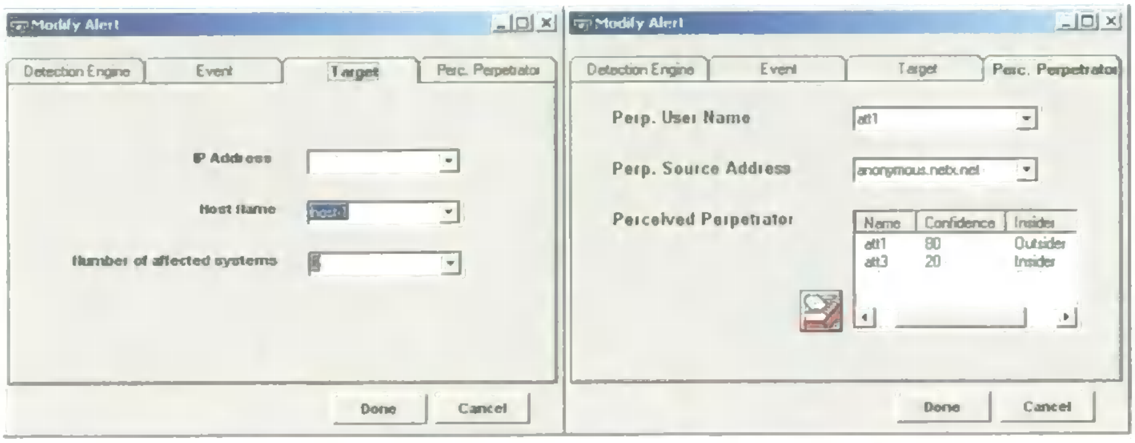
The other parameters included in each alert are the ones that the Detection Engine should provide to the Responder. These parameters can be adjusted from the console interface, as illustrated in Figures 7.4 and 7.5. The Responder can use the parameters included in the alert message as a starting point to form a decision. It should be noted that some parameters could be left blank. For example, information about a User Account in the Event tab (Figure 7.4), or the perceived perpetrator might be left empty, as they may not be relevant or available at the time of the alert.



Intrusion Confidence (0-100 %): [illegible]
Detection Source: athena.domain.gr, 195.124.23
Detection Efficiency (0-100 %): [illegible]
Alert Status (0-100): [illegible]

Time Incident Occurred: 15/05/2004 15:35:07
Attack Id: 2. Successful User Login, Authentication
Incident Id: 1
User Account:

Figure 7.4 Alert Simulation Console – Alert Parameters (1)



IP Address:
Host Name: host-1
Number of affected systems: [illegible]

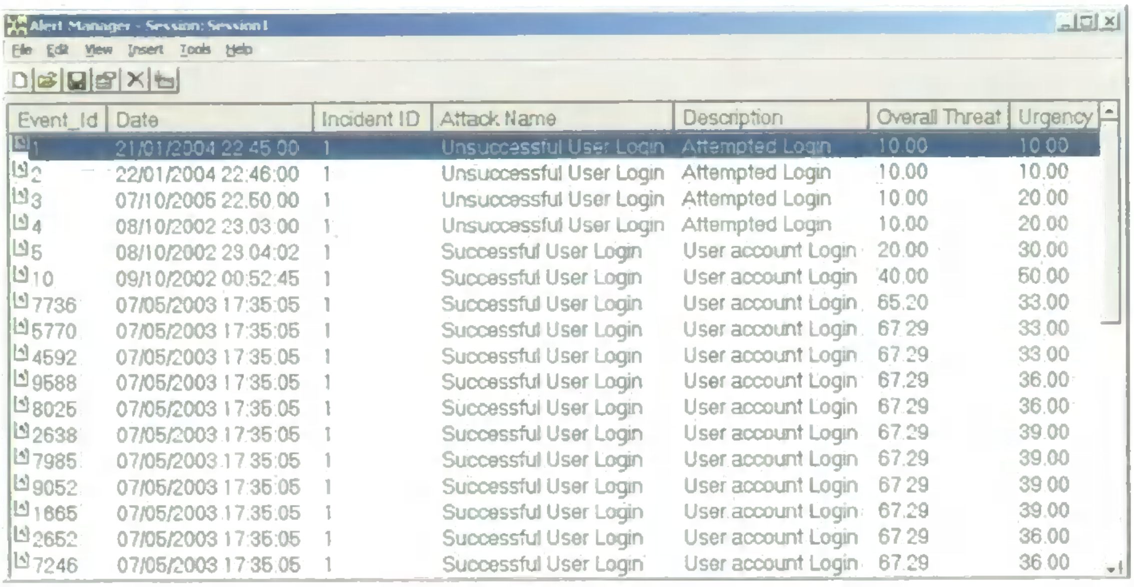
Perp. User Name: att1
Perp. Source Address: anonymous.netx.net
Perceived Perpetrator

Name	Confidence	Insider
att1	80	Outsider
att3	20	Insider

Figure 7.5 Alert Simulation Console – Alert Parameters (2)

7.4 Responder

The Responder is responsible for receiving the alerts and making response decisions according to the given context. Figure 7.6 depicts the main interface of the Responder, which logs and lists the details of alerts that have been received within a session, so that they can be tracked and reviewed by the system administrator.

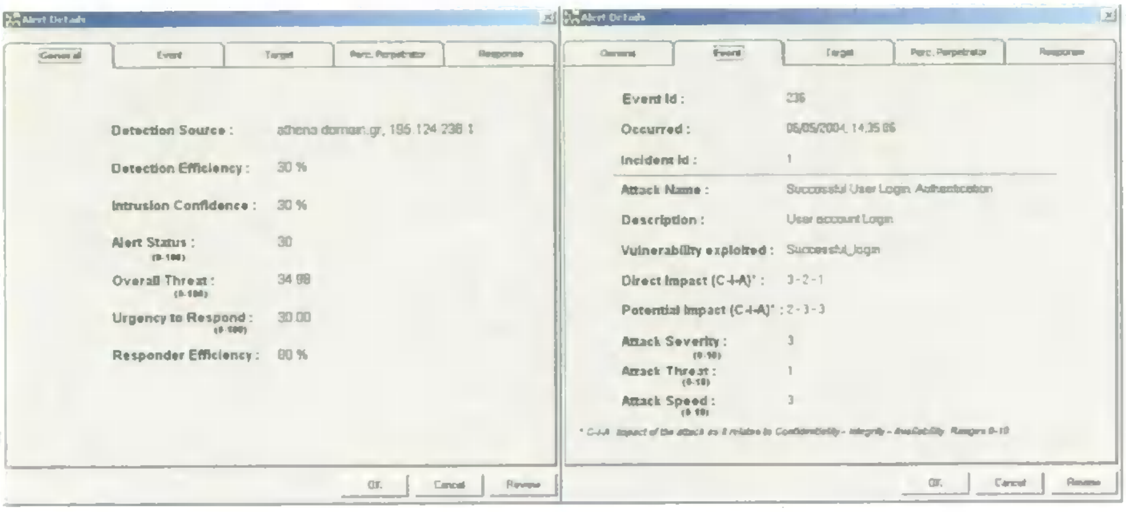

Alert Manager - Session: Session1
File Edit View Insert Tools Help

Event_Id	Date	Incident ID	Attack Name	Description	Overall Threat	Urgency
1	21/01/2004 22:45:00	1	Unsuccessful User Login	Attempted Login	10.00	10.00
2	22/01/2004 22:46:00	1	Unsuccessful User Login	Attempted Login	10.00	10.00
3	07/10/2005 22.50.00	1	Unsuccessful User Login	Attempted Login	10.00	20.00
4	08/10/2002 23.03:00	1	Unsuccessful User Login	Attempted Login	10.00	20.00
5	08/10/2002 23:04:02	1	Successful User Login	User account Login	20.00	30.00
10	09/10/2002 00:52:45	1	Successful User Login	User account Login	40.00	50.00
7736	07/05/2003 17:35:05	1	Successful User Login	User account Login	65.20	33.00
5770	07/05/2003 17:35:05	1	Successful User Login	User account Login	67.29	33.00
4592	07/05/2003 17:35:05	1	Successful User Login	User account Login	67.29	33.00
9588	07/05/2003 17:35:05	1	Successful User Login	User account Login	67.29	36.00
8025	07/05/2003 17:35:05	1	Successful User Login	User account Login	67.29	36.00
2638	07/05/2003 17:35:05	1	Successful User Login	User account Login	67.29	39.00
7985	07/05/2003 17:35:05	1	Successful User Login	User account Login	67.29	39.00
9052	07/05/2003 17:35:05	1	Successful User Login	User account Login	67.29	39.00
1665	07/05/2003 17:35:05	1	Successful User Login	User account Login	67.29	39.00
2652	07/05/2003 17:35:05	1	Successful User Login	User account Login	67.29	36.00
7246	07/05/2003 17:35:05	1	Successful User Login	User account Login	67.29	36.00

Figure 7.6 FAIR Prototype: Alert Manager

Each entry contains information about the alert itself, and the reasoning for the associated response decision. When viewing the alerts, it is also possible for the administrator to review the response decision that was made, and customise the response policy accordingly. Also, additional feedback about the effectiveness of the issued responses can be given, to inform the Responder if and / or why they were unsuccessful. A full implementation of the Responder would use this feedback as the basis for automatic refinement of the Response Efficiency and Responder Efficiency metrics, as well as the response policy. A detailed description of each alert, including details of the Responder's associated decision, can be obtained by double-clicking on it (see Figures 7.7, 7.8, and 7.9).

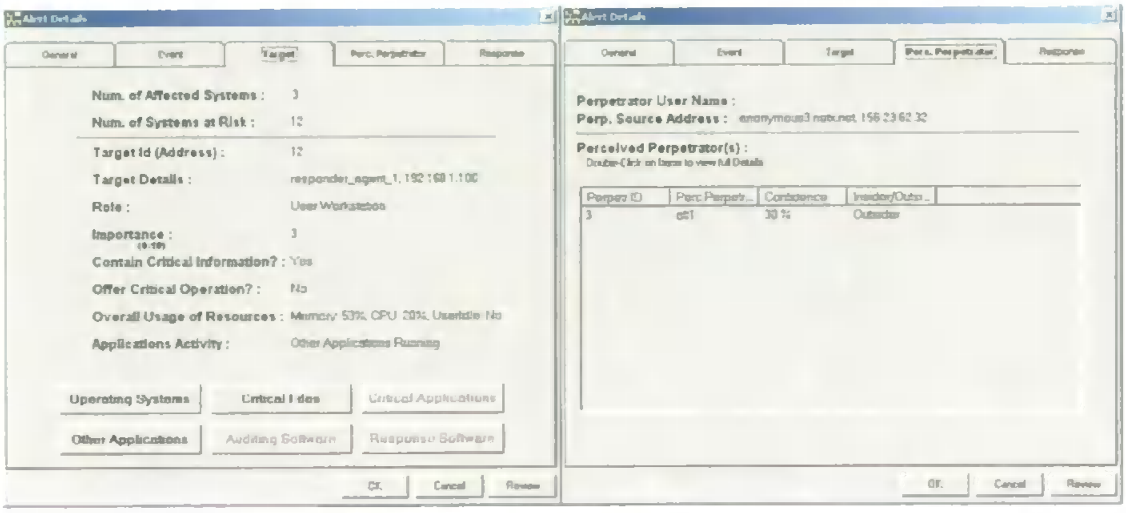
As depicted in Figure 7.8, the administrator can click on the available buttons (for example the one entitled 'Operating Systems') to view the list of operating systems or applications installed on the target. Also, by double-clicking on each suspicious perpetrator, a separate window detailing his associated characteristics will be displayed (Figure 7.10). Finally, the administrator can view more information about the characteristics of responses, by double-clicking on them, as they appear on either the list of Approved, or Candidate Responses (Figure 7.9). When the characteristics of an 'Approved Response' are displayed, the user can provide feedback about the results of the action in the form of comments such as 'Successful', 'Too severe', 'Not severe enough', and so on (Figure 7.11).



Detection Source : athena domain.gr, 195.124.238.1
Detection Efficiency : 30 %
Intrusion Confidence : 30 %
Alert Status : 30
(0-100)
Overall Threat : 34.98
(0-100)
Urgency to Respond : 30.00
(0-100)
Responder Efficiency : 80 %

Event Id : 236
Occurred : 06/05/2004, 14:35:06
Incident Id : 1
Attack Name : Successful User Login Authentication
Description : User account Login
Vulnerability exploited : Successful_login
Direct Impact (C-I-A)* : 3 - 2 - 1
Potential Impact (C-I-A)* : 2 - 3 - 3
Attack Severity : 3
(0-10)
Attack Threat : 1
(0-10)
Attack Speed : 3
(0-10)
* C-I-A: Impact of the attack as it relates to Confidentiality - Integrity - Availability. Ranges 0-10

Figure 7.7 Alert Details (1)

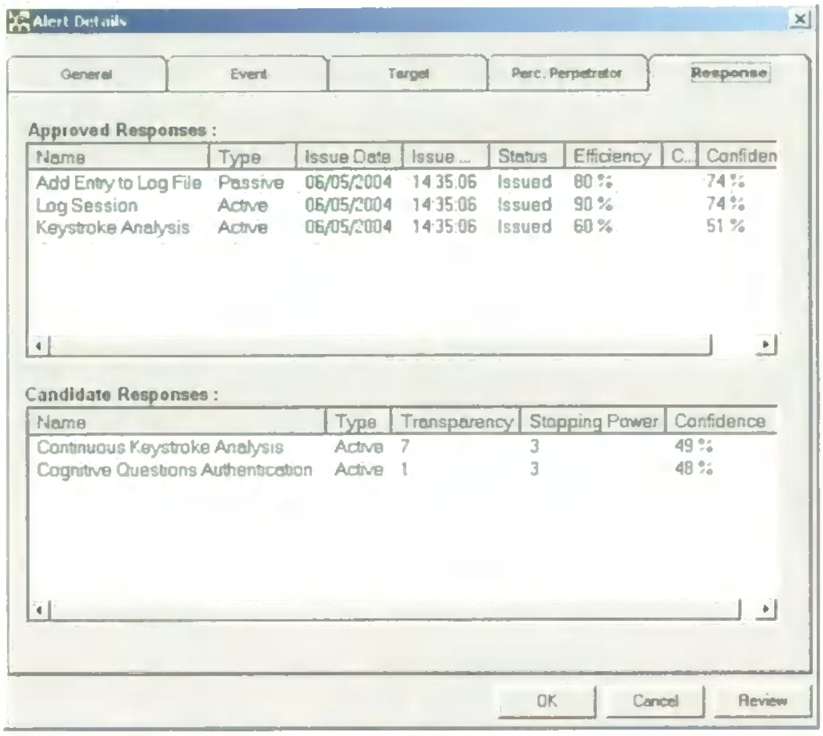


Num. of Affected Systems : 3
Num. of Systems at Risk : 12
Target Id (Address) : 12
Target Details : responder_agent_1, 192.168.1.100
Role : User Workstation
Importance : 3
(0-10)
Contain Critical Information? : Yes
Offer Critical Operation? : No
Overall Usage of Resources : Memory 53%, CPU 20%, UserIdle: No
Applications Activity : Other Applications Running

Perpetrator User Name :
Perp. Source Address : anonymous3.natx.net, 156.23.62.32
Perceived Perpetrator(s) :
Double-Click on Items to view full Details

Perpet ID	Perc. Perpetr...	Confidence	Insider/Outsi...
3	ot1	30 %	Outsider

Figure 7.8 Alert Details (2)



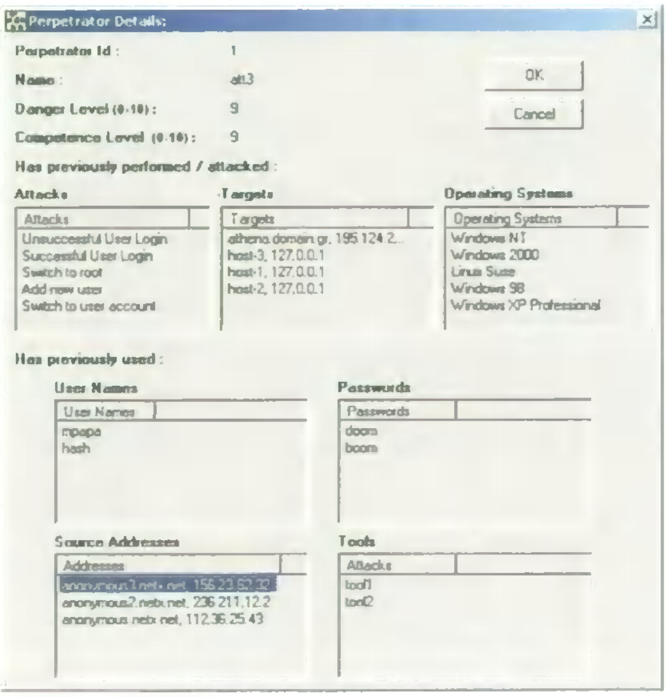
Approved Responses :

Name	Type	Issue Date	Issue ...	Status	Efficiency	C...	Confiden
Add Entry to Log File	Passive	06/05/2004	14:35:06	Issued	80 %		74 %
Log Session	Active	06/05/2004	14:35:06	Issued	90 %		74 %
Keystroke Analysis	Active	06/05/2004	14:35:06	Issued	60 %		51 %

Candidate Responses :

Name	Type	Transparency	Stopping Power	Confidence
Continuous Keystroke Analysis	Active	7	3	49 %
Cognitive Questions Authentication	Active	1	3	48 %

Figure 7.9 Alert Details (3)



Perpetrator Id : 1

Name : att3

Danger Level (0-10) : 9

Competence Level (0-10) : 9

Has previously performed / attacked :

Attacks

Attacks
Unsuccessful User Login
Successful User Login
Switch to root
Add new user
Switch to user account

Targets

Targets
athena.domain.gr, 195.124.2...
host-3, 127.0.0.1
host-1, 127.0.0.1
host-2, 127.0.0.1

Operating Systems

Operating Systems
Windows NT
Windows 2000
Linux Suse
Windows 98
Windows XP Professional

Has previously used :

User Names

User Names
mpapa
hash

Passwords

Passwords
doom
boom

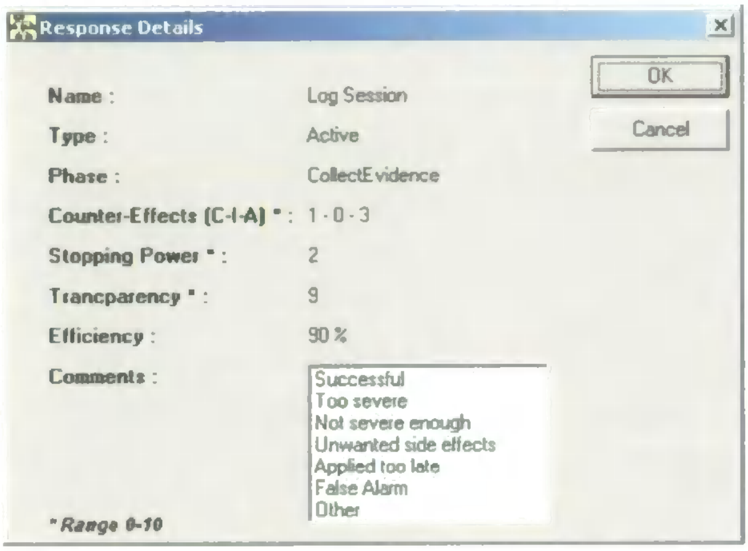
Source Addresses

Addresses
anonymous1.netx.net, 156.23.62.32
anonymous2.netx.net, 236.211.12.2
anonymous.netx.net, 112.36.25.43

Tools

Attacks
tool1
tool2

Figure 7.10 Perpetrator Details

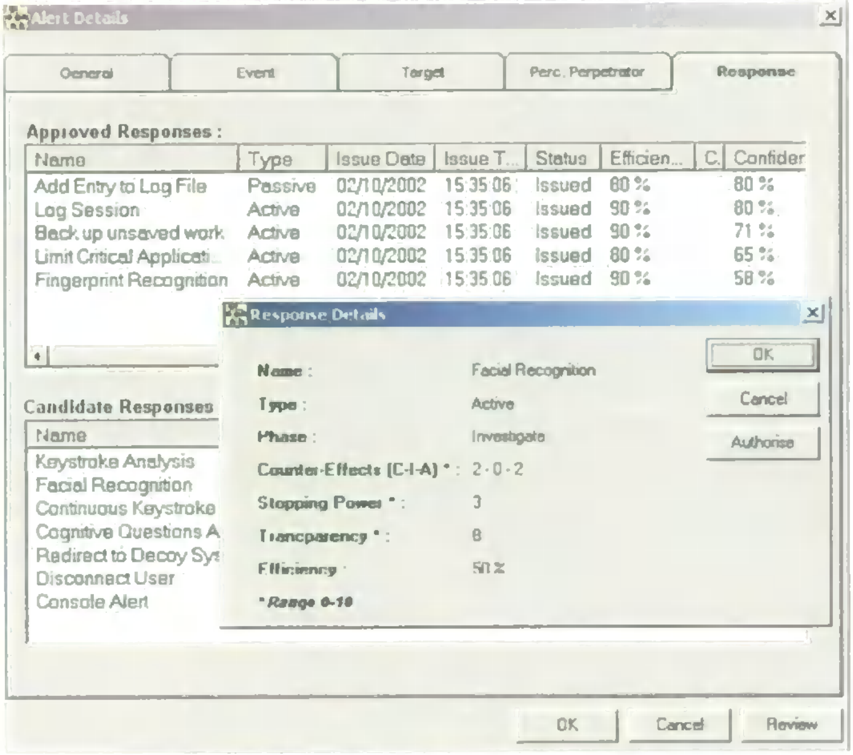


Name : Log Session
Type : Active
Phase : CollectEvidence
Counter-Effects (C-I-A) * : 1 - 0 - 3
Stopping Power * : 2
Trancparency * : 9
Efficiency : 90 %
Comments :

* Range 0-10

Figure 7.11 Issued Response Details

When the characteristics of a 'Candidate Response' are displayed, the user can authorise that response by selecting the 'Authorise' option (Figure 7.12). As already mentioned in chapter 6, the candidate response will then be issued by the Responder, and be added to the list of 'Approved Responses'. This addition does not necessarily mean that the response will be automatically approved in a future occasion. Two aspects can change that; either the customisation of the Response Policy by the administrator, or the improvement of the Response Efficiency. The latter requires establishing the outcome of the response every time it is issued, as well as the use of machine learning algorithms to automatically infer which changes on the Response Policy would most appropriately reflect this experience. The current version of the FAIR prototype does not support this feature.



Approved Responses :

Name	Type	Issue Date	Issue T...	Status	Efficien...	C.	Confider
Add Entry to Log File	Passive	02/10/2002	15:35:06	Issued	80 %		80 %
Log Session	Active	02/10/2002	15:35:06	Issued	90 %		80 %
Back up unsaved work	Active	02/10/2002	15:35:06	Issued	90 %		71 %
Limit Critical Applicati...	Active	02/10/2002	15:35:06	Issued	80 %		65 %
Fingerprint Recognition	Active	02/10/2002	15:35:06	Issued	90 %		58 %

Candidate Responses

Name
Keystroke Analysis
Facial Recognition
Continuous Keystroke
Cognitive Questions A
Redirect to Decoy Sys
Disconnect User
Console Alert

Response Details

Name : Facial Recognition

Type : Active

Phase : Investigate

Counter-Effects (C-I-A) * : 2-0-2

Stopping Power * : 3

Trancparency * : 8

Efficiency : 50 %

*Range 0-10

Figure 7.12 Candidate Response Details

After the Responder receives an alert, it tries to assess the static and dynamic context of the attack, as described in chapter 6. The static context is retrieved from a database, which contains information about characteristics of attacks, targets, attackers, applications, vulnerabilities, and so on. The Responder also assesses the dynamic context, which mainly involves the determination of the Overall Threat, the Urgency, and the characteristics of the target at the time. The last aspect in particular requires a communication with the Responder Agent, which resides at the target.

7.5 Responder Agent

The presence of the Responder Agent (RA) is not designed to be noticeable to the user. Hence, apart from the cases when it is required to interact with the user (e.g. by issuing an explicit authentication request), the RA is apparent only by the presence of an icon at the taskbar. Its role is to sit at the monitored system and wait for requests from the Responder. After the RA performs the task requested, it sends a reply to the Responder, containing the outcome of the request. The tasks supported by the RA in the prototype system are listed in Table 7.1.

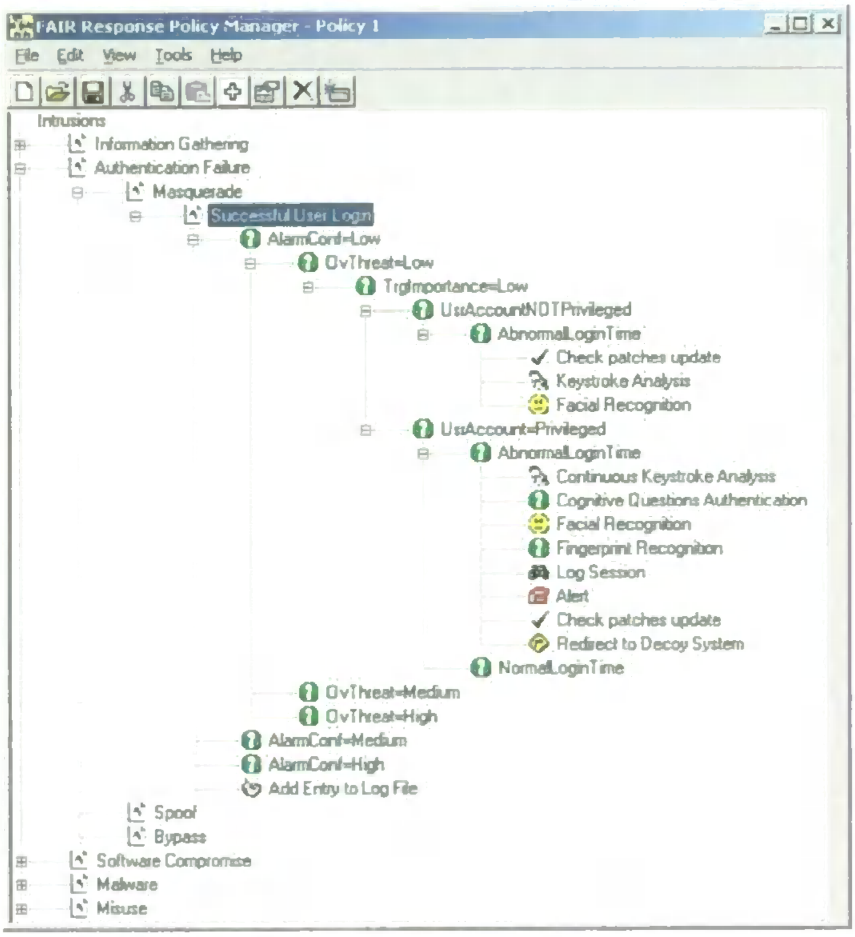
Collection	Response
Calculate the memory usage	Save status of critical applications
Calculate the CPU usage	Back up confidential files
Determine if the user is active	Terminate applications
Determine the running processes	Restrict user access to critical applications
Determine if critical applications are running	Display warning message to suspected user
Determine if critical files are open	Disconnect user
Determine if highly vulnerable (Other) applications are running	Shut down system
Determine if auditing applications are running	

Table 7.1 Tasks supported by the Responder Agent

Determining if confidential files are open, and backing them up, are tasks that are specific to the applications used to open them. Supporting the entire range of files within a system would be a very involved process. Initially the registry would need to be accessed to determine which application is used to open this type of file. Then, there should be different routine for each application, in order to perform the desired tasks. Such a process would only add complexity to the development of the Responder Agent, without adding much value to the purpose of the prototype system. Thus, only a subset of files is supported for demonstration purposes, namely files for Microsoft Word, and Microsoft Excel.

7.6 Response Policy Manager

An important element of the decision process is the Response Policy, which can be accessed and reviewed from the Responder, by selecting to view the Response Policy Manager (RPM) tool, or by clicking the 'Review' button at the Alert Details window (Figure 7.9). The RPM provides a user-friendly interface for the review of policy rules, which are represented via a hierarchical tree, where the types of alerts are at the highest level and the response actions lie at the lowest levels. The RPM allows the use of intermediate branches in the tree, which comprise the conditions under which specific response actions are initiated for particular alerts.



- Intrusions
  - Information Gathering
  - Authentication Failure
    - Masquerade
      - Successful User Login
        - AlarmConf=Low
          - OvThreat=Low
            - TrgImportance=Low
              - UsAccountNOTPrivileged
                - AbnormalLoginTime
                  - ✓ Check patches update
                  - Keystroke Analysis
                  - Facial Recognition
              - UsAccount=Privileged
                - AbnormalLoginTime
                  - Continuous Keystroke Analysis
                  - Cognitive Questions Authentication
                  - Facial Recognition
                  - Fingerprint Recognition
                  - Log Session
                  - Alert
                  - ✓ Check patches update
                  - Redirect to Decoy System
                - NormalLoginTime
          - OvThreat=Medium
          - OvThreat=High
        - AlarmConf=Medium
        - AlarmConf=High
        - Add Entry to Log File
    - Spoof
    - Bypass
  - Software Compromise
  - Malware
  - Misuse

Figure 7.13 FAIR Response Policy Manager

The Response Policy Manager is illustrated in Figure 7.13, with an illustrative example of the response rules that could be specified in relation to an 'Authentication Failure' alert. In this case, had there been an alarm from the Detection Engine describing the successful login of a suspected masquerador, the Responder would consider checking for the most recent update of related software to ensure that it is not vulnerable, initiating keystroke analysis and facial recognition (if available) to authenticate the user in a non-intrusive manner. The conditions for the latter to happen would not be just the occurrence of the incident. Only the addition of the alarm to a log file would happen in that case. For the previously mentioned responses to be issued, then the policy rules in Figure 7.13 require that the alarm confidence is low (hence the Responder would need to collect more information about the incident), and the overall threat and the importance of the target should be low as well, not justifying the issue of more severe responses. Also, the account involved would need to be not privileged, with login time outside the normal pattern, in order to issue non-intrusive authentication. If a privileged account is involved, then the Policy in Figure 7.13 shows a more elaborate range of responses being suggested as a consequence.

7.6.1 Using Policies

The RPM allows the creation of new and modification of existing response policies. In this way, different 'model' policies can be readily available as templates to either reflect different security requirements, or just simplify the task of customising them. For instance, there could be policies providing low, medium, high levels of security, or ones to reflect the changing needs of organisations. Each response policy can be saved in a database, but it is only used by the Responder to make decisions after it is applied. The latter involves opening the policy from the RPM interface and clicking the 'Apply Policy' button. That will trigger the RPM to generate the rules corresponding to the displayed tree structure, and update the expert system file, which contains them.

7.6.2 Creating and Modifying a Policy

Creating, or modifying a policy involves the addition, edit, and removal of nodes from the tree structure. Once a new policy is created, the tree structure of available alerts is automatically loaded, as illustrated in Figure 7.14.

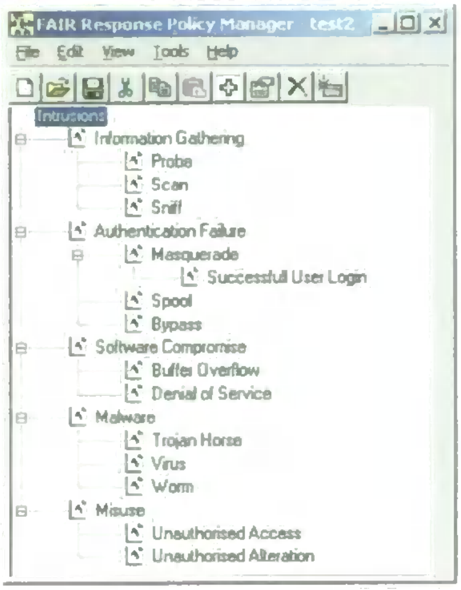
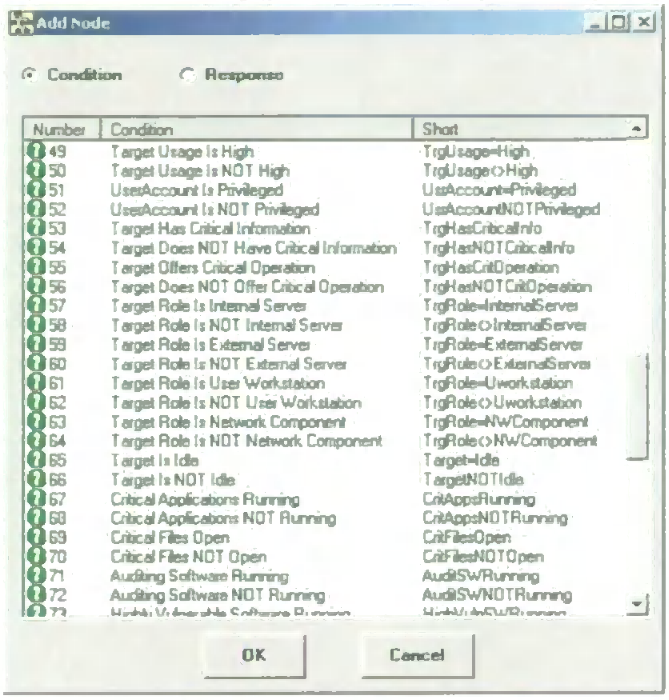


Figure 7.14 New Policy

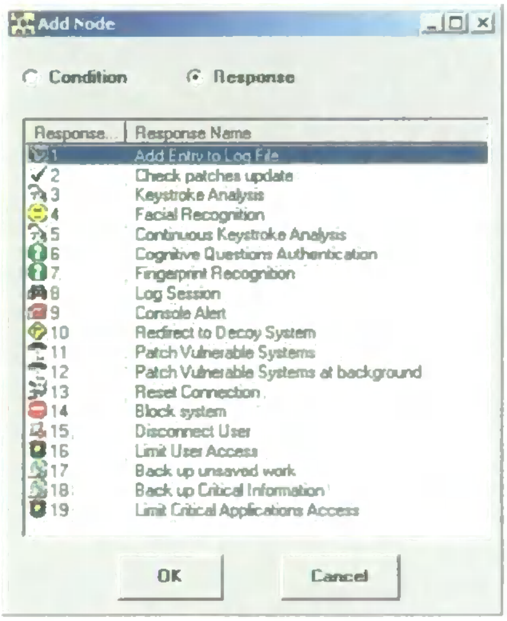
The addition, or edit, of nodes can be performed via the window provided in Figure 7.15 and Figure 7.16. Having already loaded the available alerts, each node added or edited can either be a 'Condition', or a 'Response'. If the node is added, its position will be as a child of the selected node in the tree structure (in Figure 7.14 the selected node is entitled 'Intrusions'). In addition, it is possible to Cut, Copy and Paste nodes (and their children), in order to facilitate the customisation of response policies even further.



● Condition ○ Response

Number	Condition	Short
49	Target Usage Is High	TrgUsage=High
50	Target Usage Is NOT High	TrgUsage<>High
51	UserAccount Is Privileged	UsrAccount=Privileged
52	UserAccount Is NOT Privileged	UsrAccountNOTPrivileged
53	Target Has Critical Information	TrgHasCriticalInfo
54	Target Does NOT Have Critical Information	TrgHasNOTCriticalInfo
55	Target Offers Critical Operation	TrgHasCritOperation
56	Target Does NOT Offer Critical Operation	TrgHasNOTCritOperation
57	Target Role Is Internal Server	TrgRole=InternalServer
58	Target Role Is NOT Internal Server	TrgRole<>InternalServer
59	Target Role Is External Server	TrgRole=ExternalServer
60	Target Role Is NOT External Server	TrgRole<>ExternalServer
61	Target Role Is User Workstation	TrgRole=Uworkstation
62	Target Role Is NOT User Workstation	TrgRole<>Uworkstation
63	Target Role Is Network Component	TrgRole=NWComponent
64	Target Role Is NOT Network Component	TrgRole<>NWComponent
65	Target Is Idle	Target=Idle
66	Target Is NOT Idle	TargetNOTIdle
67	Critical Applications Running	CritAppsRunning
68	Critical Applications NOT Running	CritAppsNOTRunning
69	Critical Files Open	CritFilesOpen
70	Critical Files NOT Open	CritFilesNOTOpen
71	Auditing Software Running	AuditSWRunning
72	Auditing Software NOT Running	AuditSWNOTRunning
73	Highly Vulnerable Software Running	HighVulnSWRunning

Figure 7.15 Adding a Condition



○ Condition ● Response

Response...	Response Name
1	Add Entry to Log File
2	Check patches update
3	Keystroke Analysis
4	Facial Recognition
5	Continuous Keystroke Analysis
6	Cognitive Questions Authentication
7	Fingerprint Recognition
8	Log Session
9	Console Alert
10	Redirect to Decoy System
11	Patch Vulnerable Systems
12	Patch Vulnerable Systems at background
13	Reset Connection
14	Block system
15	Disconnect User
16	Limit User Access
17	Back up unsaved work
18	Back up Critical Information
19	Limit Critical Applications Access

Figure 7.16 Adding a Response

7.6.3 Applying a Policy

As already mentioned, it is possible to view and modify different policies from the RPM interface, but they are only used by the Responder to make decisions, after they are applied. The process of applying a response policy is initiated after clicking the 'Apply Policy' option, and is summarised in Table 7.2.

1. Each node in the tree structure can either be an Alert, a Condition, or a Response. The RPM can distinguish whether a node is one or the other.

2. For each Alert, Condition, or Response, a string is pre-defined – also referred to as 'CLIPS string' – with which the node is represented in CLIPS. So, for example, the CLIPS string with which the alert 'Successful User Login' is represented, is 'successful-user-login'. The CLIPS string for the condition 'Target is Vulnerable' is 'target-vulnerable is yes'.

3. When the user initiates the process, a recursive algorithm is used to navigate (scan) every node in the tree structure.

    i) As soon as a Response node (R_i) is found, the tree is navigated upwards, in order to enumerate the Conditions separating the Response from the Alert. The Conditions are all the parents of the Response in the tree, up to the point when an Alert node is found. The CLIPS string for each Condition is stored in a table.

ii) As soon as an Alert is found, the number of conditions (N_c) is determined. Also, a certainty value is calculated, by dividing 100 with N_c.

iii) The last part of a rule is created first, by determining which responses are recommended. A string (S_R) is created for the Response R_i and all its sibling responses in the tree. Assuming that the CLIPS string of each response is R_{CLIPS}, the string S_R will contain a line for each response: “best-name is 'R_{CLIPS}' with certainty '100/N_c'”. For example, if there were 5 conditions, and 2 sibling responses to R_i, the string S_R would be:

```
(then  best-name is check-patches-update with certainty 20 and
       best-name is keystroke-analysis with certainty 20 and
       best-name is facial-recognition with certainty 20))
```

iv) Then, the first part of the rule is constructed, by defining the conditions, under which the responses are recommended. A string (S_{Ci}) is created for each Condition to construct a CLIPS rule, using the CLIPS string of the Condition, the CLIPS string of the Alert, and the string S_R. For example, S_{Ci} for the Alert ‘successful-user-login’, and the Condition ‘privileged-user-account is no’ could be:

```
(rule (if alarm is successful-user-login and privileged-user-account
      is no)
      (then  best-name is check-patches-update with certainty 20 and
            best-name is keystroke-analysis with certainty 20 and
            best-name is facial-recognition with certainty 20))
```

v) The strings for each Condition are concatenated together. For example, if the number of conditions were 5, the final string would contain 5 rules, each connecting the Alert, one of the 5 Conditions, and the string S_R.

vi) Focus of the tree scanning process moves to the last sibling of Response R_i, which will enable the recursive navigation of the tree to continue.

4. When all the nodes of the tree have been scanned, the final string (S_{Rules}) contains all the CLIPS rules responsible for the initial response selection phase, with which candidate responses are determined (see section 6.5).

5. CLIPS code is contained in a text file, within which the string S_{Rules} should be positioned. Its position is neither at the beginning, nor at the end of that file, and thus the simpler way to reconstruct the CLIPS file and update the rules is to divide it in three parts. The 2nd part is the S_{Rules} and the 1st and 3rd are the parts preceding and following it. These parts are stored in two different text files, and after they are opened, they are concatenated with S_{Rules} to reconstruct and overwrite the CLIPS file.

Table 7.2 Applying a Response Policy

7.7 Demonstrating the FAIR prototype

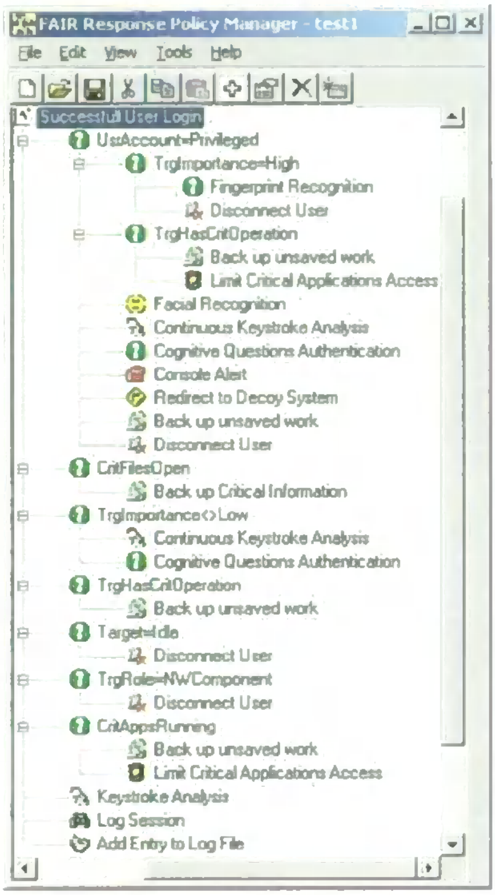
Having described the role and main features of the prototype system, this section will provide examples of how different contextual factors can influence the response decision process, and how the Responder can adapt its decisions to reflect changes in the environment. Specifically, the examples presented demonstrate how the occurrence of the same incident can trigger different responses in different contexts, thus proving the flexibility of the new approach. The occurring incident, in all examples, is a "Successful User Login" alarm, and the context in which it is occurring reflects low, medium, and high levels of severity. Finally, it should be noted, that the Responder Efficiency has been configured in all cases to be very high (90%), enabling the Responder to issue severe responses, when needed. All of the examples can be demonstrated in practice on the prototype system.

7.7.1 Alarm description

The "Successful User Login" alarm can be created as result of a suspicious user login. Some of the reasons a user login might be flagged as suspicious are the following:

- Abnormal login time;
- Abnormal login source location;
- Previous unsuccessful login attempts;
- Already suspicious login username, as a result of it being associated with other security incidents;
- Target already associated with other security incidents.

The occurrence of this alarm in current IDS systems would normally result in a log entry, which would be used for future reference (Internet Security Systems 2001). Alternatively, for more severe cases, the administrator would be alerted, or the user disconnected. According to the Response Policy used for these examples, the Responder of the FAIR prototype should consider adding an entry to a log file, log the suspicious session, perform keystroke analysis, issue cognitive questions, perform facial recognition, back up critical applications, limit access to critical applications, disconnect the user, redirect to a decoy system, and so on. The full Response Policy is illustrated in Figure 7.17 in more detail.



```
Successfull User Login
    UsrAccount=Privileged
        TrgImportance=High
            Fingerprint Recognition
            Disconnect User
        TrgHasCritOperation
            Back up unsaved work
            Limit Critical Applications Access
        Facial Recognition
        Continuous Keystroke Analysis
        Cognitive Questions Authentication
        Console Alert
        Redirect to Decoy System
        Back up unsaved work
        Disconnect User
    CritFilesOpen
        Back up Critical Information
    TrgImportance<>Low
        Continuous Keystroke Analysis
        Cognitive Questions Authentication
    TrgHasCritOperation
        Back up unsaved work
    Target=Idle
        Disconnect User
    TrgRole=NWComponent
        Disconnect User
    CritAppsRunning
        Back up unsaved work
        Limit Critical Applications Access
    Keystroke Analysis
    Log Session
    Add Entry to Log File
```

Figure 7.17 Response Policy

7.7.2 Case Example 1 – Low severity

In the first case, the Alert Simulation Console sends an alert to the Responder, with the characteristics included in Table 7.3. Once the Responder receives the alert, it will retrieve additional information about the target 'Responder_Agent_1' and the perpetrator 'Att1', which is summarised in Table 7.4 and Table 7.5.

Alert Details	
Intrusion confidence	30 (out of 100)
Detection Efficiency	30 (out of 100)
Alert Status	30 (out of 100)
Incident ID	1
User Account	Username-1 (not privileged)
Target	Responder_Agent_1
Number of Affected Systems	3 (out of 14 systems in the network overall)
Perpetrator User Name	
Perpetrator Source Address	anonymous3.netx.net
Perceived Perpetrator	Att1
Perceived Perpetrator Confidence	30
Insider/Outsider?	Outsider

Table 7.3 Alert 1 details

Target Details	
Role	User Workstation
Importance	3 (out of 10)
Dependant Systems	1 (out of 14 systems in the network)
Operating Systems	Windows XP
Critical Information?	Yes (File Location)
Critical Operation?	No
Other Applications	Windows Media Player

Table 7.4 Target 'Responder_Agent_1' details

Attacker Details	
Danger	3 (out of 10)
Competence Level	3 (out of 10)
Attacks Performed	Low number, Low severity
Operating Systems Attacked	Windows XP (the OS of the target)
Targets Attacked	Low number: Not the target
Aliases	"ask" (not detected in the target)

Table 7.5 Attacker 'Att1' details

The Responder also assesses the dynamic context of the attack, including the current status of the target. The dynamic context is summarised in Table 7.6.

Dynamic Context	
Alarm Confidence	9 (out of 100)
Perpetrator Threat	8.85 (out of 100)
Overall Threat	34.98 (out of 100)
Urgency	30 (out of 100)
Number of Systems at Risk	11 (out of 14)
Memory Usage at target	52 (out of 100)
CPU Usage at target	20 (out of 100)
Target Idle?	Yes
Critical Applications Running?	No
Critical Files Open?	No
Other Applications Running?	Yes
Auditing Software Running?	No

Table 7.6 Dynamic Context: Alert 1

The level of suspicion for the occurrence of an attack is very low, as the Alarm Confidence suggests (9%). Also, the suspected perpetrator does not seem to represent a significant danger overall, as the Perpetrator Threat is only 8.85%. Combining these metrics with the fact that most of the contextual factors suggest an attack with low severity, against a user workstation, with low importance, explains why the Overall Threat of the event is estimated at relatively low levels (34.98%). In fact, the only aspects responsible for the slight increase of the Overall Threat from the level of 30% are the facts that the target has critical information and highly vulnerable software, which was running at the time of the alert. Also, the fact that the Number of Systems at Risk is relatively high was another factor causing the Overall Threat to increase. The speed of the attack is low (3 out of 10), and thus the Urgency to Respond is low as well (see section 6.4.6.2).

Based upon these factors and the Response Policy (see Figure 7.17), the Responder approved 3 responses, which are listed in Table 7.7.

	Response	Confidence	Counter-Effects (max 30)	Stopping Power (max 10)	Transparency (max 10)	Efficiency
Approved Responses	Add Entry to Log File	74%	1	0	9	80%
	Log Session	74%	4	2	9	90%
	Keystroke Analysis	51%	6	3	7	60%
Suggested Responses	Continuous Keystroke Analysis	49%	10	3	7	80%
	Cognitive Questions Authentication	48%	7	3	1	70%
	Disconnect User	10%	13	9	1	95%

Table 7.7 Selected Responses: Alert 1

All the Approved Responses were suggested in the Response Policy, without any conditions, and thus they were expected to be candidates, regardless of any contextual factors. The reason the first two responses have been approved with higher confidence (74% instead of 51%) can be attributed to their characteristics. Keystroke Analysis has lower confidence score, since it has slightly higher Counter-Effects, and lower Transparency and Efficiency.

Apart from the Approved Responses, weaker choices were included in the list of Suggested Responses (Table 7.7). Although the first two Suggested Responses were not explicitly indicated by the Response Policy, they still received high enough score to be considered as candidates. The reason for that is that they partially satisfied some of the conditions, with which they were recommended. Most importantly, though, their characteristics were suitable enough for them to receive confidence rating of 49% and 48% respectively. The main reason for that is the fact that their phases are 'investigate', which receives high confidence rating when the Alarm Confidence is low. As for the last Suggested Response, the 'Disconnect User', although it was explicitly suggested by the Response Policy (the condition 'Target is Idle' was satisfied), its characteristics did not receive high enough matching confidence. The main reason for that was its stopping power, which was too high for an alert with low Alert Status, Alarm Confidence, Urgency, and so on (see section 6.4.7.4).

7.7.3 Case Example 2 – Medium severity

In the second example, the Alert Simulation Console reports the same attack to the Responder, but in a more severe context. The characteristics of the attack are the same, but the Intrusion Confidence, Detection Efficiency are higher (60%), the target is different (an external server with critical applications, but no critical information) and the perpetrator is more dangerous. More details about the alert, the target, and the perpetrator are provided in Table 7.8, Table 7.9, and Table 7.10 respectively. The Dynamic Context of the attack is listed in Table 7.11.

Alert Details	
Intrusion confidence	60 (out of 100)
Detection Efficiency	60 (out of 100)
Alert Status	60 (out of 100)
Incident ID	1
User Account	Username-2 (privileged)
Target	Responder_Agent_2
Number of Affected Systems	6 (out of 14 systems in the network overall)
Perpetrator User Name	doom
Perpetrator Source Address	Anonymous2.netx.net
Perceived Perpetrator	Att2
Perceived Perpetrator Confidence	60
Insider/Outsider?	Insider

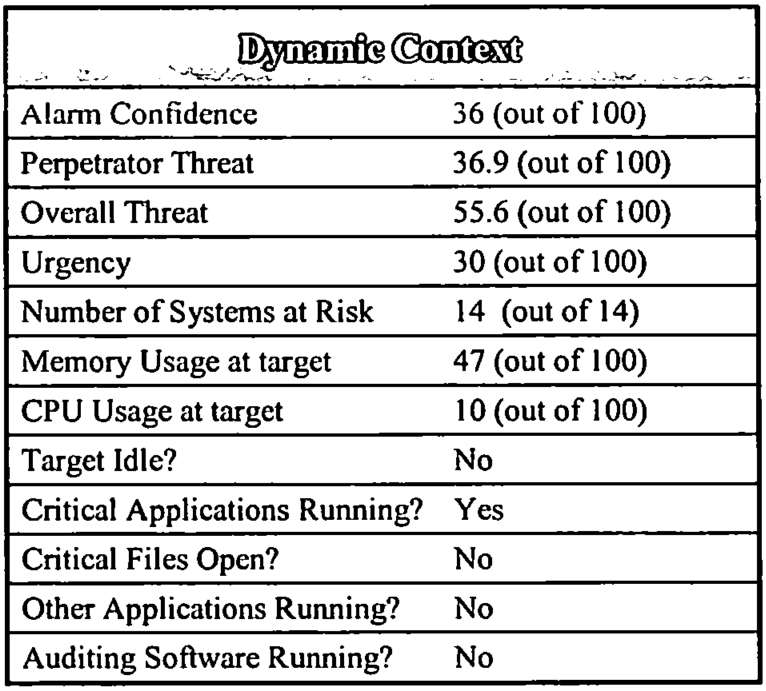
Table 7.8 Alert 2 details

Target Details	
Role	External Server
Importance	6 (out of 10)
Dependant Systems	4 (out of 14 systems in the network)
Operating Systems	Windows XP
Critical Information?	No
Critical Operation?	Yes (Microsoft Word)
Other Applications	Windows Media Player
Auditing Software	Archaeopteryx

Table 7.9 Target 'Responder_Agent_2' details

Attacker Details	
Danger	6 (out of 10)
Competence Level	6 (out of 10)
Attacks Performed	Low number, Low severity
Operating Systems Attacked	Windows XP (the OS of the target)
Targets Attacked	Low number: Not the target
Aliases	"doom" (detected in the target)

Table 7.10 Attacker 'Att2' details



Dynamic Context	
Alarm Confidence	36 (out of 100)
Perpetrator Threat	36.9 (out of 100)
Overall Threat	55.6 (out of 100)
Urgency	30 (out of 100)
Number of Systems at Risk	14 (out of 14)
Memory Usage at target	47 (out of 100)
CPU Usage at target	10 (out of 100)
Target Idle?	No
Critical Applications Running?	Yes
Critical Files Open?	No
Other Applications Running?	No
Auditing Software Running?	No

Table 7.11 Dynamic Context: Alert 2

The level of suspicion for the occurrence of an attack is higher this time, but it is still at relatively low levels, as the Alarm Confidence suggests (36%). The threat introduced by the suspected perpetrator is higher this time (36%), since he has caused considerable damage, has medium skills, but has not attacked the same target before, nor has previously perpetrated many attacks. Also, the target this time is more important, as it is an external server with higher importance and critical applications, which were running at the time. As a result, the Overall Threat is nearly twice as much as the previous alarm (55.6%). The Urgency to Respond is the same though, since it is only influenced by the speed of the attack (which is the same), and the number of alerts in the same incident.

Since the User Account was this time privileged (see Figure 7.17), a much larger number of responses were approved, or suggested. The responses selected by the Responder are listed in Table 7.12.

	Response	Confidence	Counter-Effects (max 30)	Stopping Power (max 10)	Transparency (max 10)	Efficiency
Approved Responses	Add Entry to Log File	80%	1	0	9	80%
	Log Session	80%	4	2	9	90%
	Back up unsaved work	71%	3	1	8	90%
	Limit Critical Applications Access	65%	5	5	8	80%
	Fingerprint Recognition	58%	8	3	0	90%
Suggested Responses	Keystroke Analysis	55%	6	3	7	60%
	Facial Recognition	54%	4	3	8	50%
	Continuous Keystroke Analysis	53%	10	3	7	80%
	Cognitive Questions Authentication	52%	7	3	1	70%
	Redirect to Decoy System	22%	18	7	5	50%
	Disconnect User	10%	13	9	1	95%
	Alert	9%	3	0	9	90%

Table 7.12 Selected Responses: Alert 2

The first two Approved Responses receive higher rating, as they have very low impact, and significantly high Efficiency. Backing up unsaved work at the target, involves making a copy of the content of open files (potentially unsaved work) at a specific location in the system. This aims to minimise the impact of more severe responses, especially the ones that limit or even deny user access in the system. Hence, it is particularly important to be initiated before any severe action like 'Limit Critical Applications Access', or 'Disconnect User'. In this alert, both 'Back up unsaved work', and 'Limit Critical Applications Access' have been selected as a result of the user account being privileged, the target offering critical operation, and critical applications running at the target. Also, the medium levels of Alert Status, and Overall Threat enable the use of more severe responses, with higher Stopping Power, making possible the approval of 'Limit Critical Applications Access'.

Since the Alarm Confidence is medium, and there is still scope for further investigation, responses such as Fingerprint Recognition, Keystroke Analysis, and so on, receive relatively high confidence ratings. In fact, all investigative responses receive confidence higher than 50%, which is the threshold for a response to be approved, but only the strongest candidate ultimately gets selected, since all of them aim to do the same task, which is user authentication. The reason Fingerprint Recognition is the strongest match is because it has the highest Efficiency, which is more important as the Alarm Confidence, Overall Threat, and Alert Status increase.

Responses, such as 'Redirect to Decoy System' and 'Disconnect User' still have too high Stopping Power to be approved, so they receive lower confidence ratings. Finally, the option of alerting the administrator receives low rating, as the Responder Efficiency is at a very high level (90%). Had the Responder Efficiency been lower, preventing the Responder from issuing severe responses, such as 'Limit Critical Applications Access', the confidence of alerting would have been significantly upgraded. At the same time, the confidence of limiting user access would have been significantly degraded.

7.7.4 Case Example 3 – High severity

In the third example, the context of the attack is even more severe. The characteristics of the attack are again the same, but the Intrusion Confidence, Detection Efficiency are even higher (90%), the target is an internal server with critical applications, and confidential information, and the perpetrator is even more dangerous. Full details of the alert, the target, and the perpetrator are provided in Table 7.13, Table 7.14, and Table 7.15 respectively. The Dynamic Context of the attack is listed in Table 7.16.

Alert Details	
Intrusion confidence	90 (out of 100)
Detection Efficiency	90 (out of 100)
Alert Status	90 (out of 100)
Incident ID	1
User Account	Username-2 (privileged)
Target	Responder_Agent_3
Number of Affected Systems	9 (out of 14 systems in the network overall)
Perpetrator User Name	hash
Perpetrator Source Address	Anonymous.netx.net
Perceived Perpetrator	Att3
Perceived Perpetrator Confidence	90
Insider/Outsider?	Insider

Table 7.13 Alert 3 details

Target Details	
Role	Internal Server
Importance	6 (out of 10)
Dependant Systems	9 (out of 14 systems in the network)
Operating Systems	Windows XP
Critical Information?	Yes (File Location)
Critical Operation?	Yes (Microsoft Word)
Other Applications	Windows Media Player
Auditing Software	Archaeopteryx

Table 7.14 Target 'Responder_Agent_3' details

Attacker Details	
Danger	9 (out of 10)
Competence Level	9 (out of 10)
Attacks Performed	Low number, Low severity
Operating Systems Attacked	Windows XP (the OS of the target)
Targets Attacked	High number: Not the target
Aliases	"mpapa", "hash" (detected in the target)

Table 7.15 Attacker 'Att3' details

Dynamic Context	
Alarm Confidence	81 (out of 100)
Perpetrator Threat	79.65 (out of 100)
Overall Threat	73.021 (out of 100)
Urgency	30 (out of 100)
Number of Systems at Risk	16 (out of 14)
Memory Usage at target	50 (out of 100)
CPU Usage at target	10 (out of 100)
Target Idle?	No
Critical Applications Running?	Yes
Critical Files Open?	Yes
Other Applications Running?	No
Auditing Software Running?	No

Table 7.16 Dynamic Context: Alert 3

The level of suspicion for the occurrence of an attack is at much higher level this time (81%), as the Intrusion Confidence and Detection Efficiency are very high (90%). The Perpetrator Threat is also high (79.65%), since the suspected perpetrator has high Danger and Competence levels, has attacked the same target before, and has long attack history. Also, the target this time has higher importance, as it is an internal server with critical applications, and critical information, which were both running / open at the time. As a result, the Overall Threat is at a high level (73.021%). The Urgency to Respond is again the same, since the speed of the attack has not changed, and the number of alerts in the same incident has not exceeded the threshold, which warrants an increment of the Urgency (that threshold has been defined for all examples to be 10 alerts).

The responses selected by the Responder are listed in Table 7.17.

	Response	Confidence	Counter-Effects (max 30)	Stopping Power (max 10)	Transparency (max 10)	Efficiency
Approved Responses	Add Entry to Log File	84%	1	0	9	80%
	Log Session	84%	4	2	9	90%
	Back up Critical Information	74%	3	1	8	90%
	Back up unsaved work	74%	3	1	8	90%
	Limit Critical Applications Access	68%	5	5	8	80%
	Disconnect User	58%	13	9	1	95%
Suggested Responses	Redirect to Decoy System	38%	18	7	5	50%
	Facial Recognition	11%	4	3	8	50%
	Keystroke Analysis	10%	6	3	7	60%
	Continuous Keystroke Analysis	10%	10	3	7	80%
	Alert	9%	3	0	9	90%
	Cognitive Questions Authentication	4%	7	3	1	70%
	Fingerprint Recognition	3%	8	3	0	90%

Table 7.17 Selected Responses: Alert 3

Since many contextual factors were at high levels, more severe responses were approved this time, especially ones that aim to protect the system, rather than investigate the incident, or alert the administrator. In fact, investigative responses have received low ratings, due to the fact that the Alarm Confidence is at high levels, and therefore there is no need for further investigation. Also, the fact that the Alert Status, Overall threat, Responder Efficiency have high levels, enables the Responder to issue responses with high Stopping Power. Thus, although 'Disconnect User' was indicated in the Response Policy in Figure 7.17, it is only in this case that it is actually allowed with confidence 58%. Even if 'Redirect to Decoy System' is already indicated by the Response Policy, it does not receive high enough rating to be approved, as its efficiency is not high enough. In this case, the efficiency of a response increases its priority even more, and thus the confidence ratings are adjusted to reflect the changing requirements for the selection of responses. Finally, the alerting option receives again low confidence, mainly due to the fact that the Responder Efficiency is so high. Had the Responder Efficiency been lower, not allowing the disconnection of the attacker, the Alerting option would have received much greater confidence, especially since the 'Disconnect User' response would not have been approved.

7.8 Conclusions

This chapter described the FAIR prototype system, demonstrating two main features of the proposed architecture; the ability to provide easily customisable Response Policies, and the ability to adapt decisions according to changes in the environment. The customisation of Response Policies is enabled via the user-friendly interface of the Response Policy Manager. Its main feature is that instead of requiring knowledge of expert systems rules code to modify rules in the response policy, it represents the rules in a tree structure, where alerts are at the highest levels of the tree, the responses are at the lowest levels, and the conditions under which responses should be considered are at the intermediate nodes.

The ability to adapt decisions according to changes in the environment has been demonstrated in the form of three examples. In all examples, the same alert was sent to the Responder, but with an increasingly severe context each time, with the result that response decisions were adapted accordingly.

The development of the prototype has also served to inform the FAIR architecture, especially in areas relating to how response factors are defined and used. For example, the Response Action, and the Target characteristics were modified as a result of the prototype development, in order to reflect the way they are used in the Response Policy. Overall, the prototype has served to prove the viability of the FAIR architecture, and in doing so, it has provided a practical validation of the ability to achieve flexible automated and intelligent response. In that sense, the system in its current form is believed to represent an advancement on existing intrusion response approaches.

CHAPTER 8

CONCLUSIONS

8 CONCLUSIONS

This chapter concludes the thesis by summarising the achievements and highlighting the limitations of the research. It also identifies new research areas within which the work presented could be enhanced.

8.1 Achievements of the research programme

The research programme has met all of the objectives originally specified in chapter 1, with new conceptual and practical work being undertaken in a number of areas. The specific achievements were:

1. Limitations of existing intrusion response mechanisms and the problems preventing the adoption of automated, active methods have been established (See Chapters 3 and 4). Contributions of previous research have also been reviewed, establishing the need and scope for further improvement (Chapter 4).

2. The Response-Oriented Taxonomy of Intrusions has been formulated (Chapter 5), aiming to facilitate the systematic study of intrusions and give insight into the issue of intrusion response, by focusing upon intrusion characteristics that can influence the response process. These characteristics represent the varying results of intrusions, as they are subjected to different targets. The Response-Oriented Taxonomy, unlike existing intrusion taxonomies, focuses upon detectable intrusions, and represents the results of an intrusion in multiple aspects, such as *Urgency*, *Severity*, *Impact(s)*, and *Potential Incidents*.

3. General response mechanisms, appropriate for counteracting intrusions with different characteristics have been identified (Chapter 5). The main influencing characteristics were the severity, urgency, and the phase of the attack.

4. The context of an occurring attack has been assessed and represented within a system, in a comprehensive manner (Chapter 6). A number of factors have been identified, as they relate to characteristics of intrusions, targets, legitimate users, attackers, response actions, the Detection Engine, and the Responder itself.

5. The Flexible Automated and Intelligent Responder (FAIR) Architecture has been designed and developed, as the architecture in which the above-mentioned response factors could be used to enhance the response capabilities of a system (Chapter 6). The main concepts of the proposed architecture, apart from representing the context of an attack in a more comprehensive manner, are the ability to adapt decisions according to changes in the environment, and facilitate the customisation of Response Policies.

6. A proof-of-concept prototype has been designed and realised, in order to validate the feasibility of the proposed architecture, and prove the ability to achieve flexible automated and intelligent response (Chapter 7).

Several papers relating to the research programme have been presented at refereed conferences and journals, (these are attached in Appendix C) and have received positive comments from delegates and reviewers. As such, it is believed that the research has made valid and useful contributions to the IT Security field, and specifically in the context of intrusion detection and response.

8.2 Limitations of the research

Despite having met the overall objectives of the research programme, it is nevertheless possible to identify a number of limitations associated with the work. The main limitations of this programme are listed below.

1. The ratings of the intrusion results presented in the Response-Oriented Taxonomy cannot be considered practically usable in a wider context, since they only show comparative differences between attacks, and the contexts in which they occur. Also, the cases of attacks considered were very generic, not allowing the reflection of individual characteristics of alarms within the same category. In spite of these limitations, however, it was considered that a more detailed and accurate approach would not add much value to the purpose of the taxonomy at that stage, which was the indication of generic response mechanisms.

2. The FAIR architecture would not be directly applicable within many real-world networks, as it does not support distributed environments, or heterogeneous monitored clients with varying processing capabilities. However, a simpler (centralised) model was preferred to demonstrate the concepts of the architecture more clearly, without introducing additional complexity or losing anything from the research value of the work. With appropriate adjustments, the concepts of the FAIR architecture can still be applied in a more complex model, with research

projects such as Emerald (Porras and Neumann 1997) having already demonstrated the feasibility of operation within a distributed environment.

3. The available time did not permit full implementation of the Responder and Responder Agent, especially the features related to the feedback from the administrator and adjustment of the Response Efficiency. Again, however, it was considered that devoting additional focus to these aspects would not have added significant value in the current version of the system. It would, however, become meaningful once machine-learning mechanisms were incorporated into the system. However, this aspect represents an issue for further research in its own right.

8.3 Suggestions and scope for future work

It is possible to identify a number of areas, in which future work could be conducted to build upon what was undertaken in this project. A number of ideas have already been covered in the previous chapters. These areas, together with new ones are summarised below.

1. The factors related to attacker profiles were not specified in a very detailed manner. The principal reason was that, although the information about attacker profiles was considered important for response, research has not progressed so far in this area. Thus, it would be premature to present detailed descriptions of information that have not been proven yet to be viable. Further research, though, could enable a more detailed specification and meaningful use of attacker profiles.

2. Further work could contribute to the detection and response to attackers according to their behavioural characteristics. For example, one aspect of research could be to monitor how attackers respond to specific (and sometimes unpredictable) events happening at the target, during an attack, and how effectively these events could be used as methods of response. Examples of these events are a login by the system administrator, messages from unknown applications, or warning messages about insufficient resources (memory or disk space). In addition, deception techniques could be employed in this investigation.

3. The assessment of contextual factors, especially those assessed by the Responder (Overall Threat, Urgency, Number of Systems at Risk, etc), was not based upon real incidents and alarms, and thus it was only indicative. A more accurate calculation of these values would need to be based upon real incidents and would thus involve the trial of the response mechanism in a real network.

4. The feedback mechanism could be more meaningfully advanced with the integration of machine learning algorithms. Also, the learning ability of the FAIR prototype will be greatly enhanced by automatically determining the results of an attack and a response.

5. Apart from the response mechanism, the FAIR prototype could also be used in a real network, to evaluate its performance. This process can also enable the formulation of 'model' response policies, which will be used to reflect low, medium, or high levels of security (as described in section 7.6.1).

6. The task of creating and updating System Profiles automatically, without human intervention, is another area that can significantly increase the level of automation in intrusion response. Therefore, Passive or Active Fingerprinting (as discussed in section 6.4.3) represents another research area, in which the work presented in this thesis can be enhanced.

8.4 The future for automated response

The growing dependence upon Information and Communication Technologies highlights the need for advanced security countermeasures. Intrusion Detection Systems have gained wide acceptance within the IT community, as passive monitoring tools, which report security problems to a system administrator. However, the role of IDSs needs to evolve, as networking infrastructures become larger, more complicated, and difficult to manage, resulting in overwhelming numbers of IDS alarms. Also, as the attacks become more automated, sophisticated, and can be initiated by decreasingly skilled attackers, they leave a much-reduced timeframe for response. As a result, the ability of IDSs to automatically counteract attacks is becoming increasingly important.

The adoption of automated response now represents a reality in the intrusion detection domain, with IDS vendors placing focus upon the response features of their products. Even in their current form, IDS / IPS products can contribute significantly in the protection of systems and networks, by preventing and stopping verified threats. Nonetheless, the problem has been far from solved in the commercial marketplace, and this research has contributed to the domain at several levels. It has served to highlight the importance of automated response, and contributed in the understanding of the threats by proposing the

Response-Oriented Taxonomy of Intrusions. More significantly, it has focused upon enhancing the response decision mechanism, by basing decisions not only upon the attack, but also the context in which it is occurring. As a consequence, the decisions are able to support flexible and escalating levels of response, according to that context. From a wider perspective, the customisable nature of response policies also provides a means to reflect the changing needs and characteristics of organisations.

The adoption of these approaches will enable automated response technologies to mature. This will ultimately reduce their potential to impact upon legitimate users and systems by disrupting legitimate activities, and instead offer a facility by which organisations can reliably protect their systems.

APPENDIX A

APPENDIX A

COMMENTS ABOUT INTRUSION RESPONSE

Comments from IDS vendors

"We have not published anything on what an appropriate response would be to specific attacks. Our implementation has a few beliefs built into it, but those are not documented outside the code. Those beliefs are quite simple: (1) probes deserve to be traced and reported, but no blocking rules are applied at firewalls or filtering routers; (2) floods deserve to be traced, reported, and stopped using rate limiting mechanisms in filtering routers/firewalls; and (3) other attacks deserve to be traced, reported, and stopped using packet filtering rules. We also developed a capability to attempt isolation of a machine that has been çompromised. Finally, we developed some host-based mechanisms that attempt to respond to host-based detector alerts by performing actions such as killing a user's session or disabling a compromised user account."

Dan Schnackenberg, Boeing Phantom Works

"our product at this time does not provide a particular response mechanism out of the box,... What I have found is that, at least in North America, many of the security professional prefer not to use automated response systems. For the network management world, it is relatively easy to dictate when some action is required based on hardware or software failure. In the security world we must assume that the information that is provided, is merely untrusting information, and the degree of event correlation is paramount. The more you can correlate the events the easier it is to analyse the degree of threat. So it then is very difficult to run automated response, shut down applications,

change router settings, prevent access to the Internet. I am sure that you have heard of the Hacker technique of creating "noise". It is very difficult to determine noise and the real thing. If shutting down or diverting services is the response then the hacker has won. The prime principles of confidentiality, integrity and Availability are destroyed from response type."

Bill Oliphant, Dir Product Management,

Intellitactics Inc., Network Security Manager (NSM)

"We have the functionality for automated responses but we haven't yet explored fully what we should do to pro-actively deal with suspected intrusions. Though we haven't tested the following options these are what we've come up with so far:

We can start other IDS tools to further determine if an attack is taking place. We can start other more sophisticated decision making systems and feed them log files to further identify attacks. We can alert other servers that an attack is possibly taking place."

Steve Haler, University of Idaho,

Centre for Secure and Dependable Software (CSDS)

"we do not have any real-time intrusion system to "guess" on suspected possible compromised systems."

Joshua Carlson, Technical Support Analyst, Tripwire, Inc.

"In short, there is no automated incident response mechanism built into SHADOW. Basically the issues are spoofed/anonymized source addresses and what action to take.

I've read (a posting from H.D. Moore) where at one time one could use the TTL field to determine (most times) if an IP is spoofed or not. But shortlv after, Nmap (and probably other tools) began randomising values put in TTL fields so that was no longer reliable. Plus, even if one can determine the real source, one is still left with the problem of what action to take. Retaliation (DoS for instance) is out of the question since one can't ignore the impact on innocent users coming from the same network (say with an ISP). Therefore, SHADOW will continue to only support detection and reporting."

Rob Blader, Information System Assurance Office,

CD2S, Naval Surface Warfare Center

"We do not have any self-imposed automated proactive responses; we enable the creation of policies for response. We facilitate filtering and review/analysis, along with detailed drill-downs for data for security management decisions"

Benedict M. Campbell, NetForensics Inc.

" netForensics does notification based on the severity of the events coming in. So netForensics is more of a monitoring and managing solution than an intrusion response system."

Dhani Amaratunge, Technical Marketing Analyst, NetForensics Inc.

"..Dragon does support automatic responses by allowing the end user to run programs via a tool we call 'AlarmTool'. These programs can be written by the end user or commercial programs (such as emailers, pagers, and other notification methods). Some customers do effect the configurations or rules of routers and firewalls.

We at Enterasys (Dragon IDS) do not advocate the use of this tool to change rules on routers or firewalls. We believe that you can easily cause a DOS on yourself, and are especially bad if this is what the attacker wanted in the first place. We think it is dangerous to put all your faith in automatic responses believing that you are protected. Attackers are very smart and know how to use your own equipment against you if it will benefit their attack. We supply the end user with forensics of the alert (i.e. you can replay a session, look at the raw data packet, pull in firewall logs for correlation). We believe that the ability to see if an attack was successful or not and then have a human acting on that is better for the overall health and security of the network.

We do have a tool that will shut down particular sessions, but will not block that IP from acting again. We consider this a compromise between automated response and no response at all. ...most people in the security product industry agree that automatic responses can be very dangerous and should not be relied upon to make important decisions about your networks."

Sales at Enterasys Networks (formerly Network Security Wizards),
<sales@network-defense.com>

"Currently, IntruShield products implement several strategies for assessing the result status of an attack. We reports the result status as one of the following:success, when the attack is an exploit which has obtained a remote shell on the victim; failed, when the attempt was not able to have the expected impact; unknown, when there is not sufficient info to assess the result status; suspicious, when the event is indicative of suspicious activity; blocked, when the sensor is deployed inline and the attempt has been blocked.

Failed status is relatively easier to determine. Based on full-stateful application protocol parsing, we can differentiate between application request versus response, we can recognize different implementations (e.g. apache versus iis), and we will know if an request has been accepted or rejected by the server. An attempt will fail if it's against the wrong implementation or rejected by the server.

Determination of Success status is harder. Currently, the most reliable indication will be for the exploit type of attacks, when a remote shell is obtained; Other cases include certain information disclosure attacks where corresponding info was exposed. There are other modes of success that require further correlation to cover. While the Success status can be useful for users operating in pure IDS mode, it's better to block these upfront by deploying IntruShield in prevention mode (inline blocking).

It should be noted that vulnerability information will be important in tailoring IDS/IPS policies and in further alert correlation for data reduction, real-time result-status assessment is an important "layer" for accurate detection/protection beyond the vulnerability information for the target.

Fengmin Gong, Chief Scientist,

McAfee Network Protection Solutions BU, Network Associates

Comments from Intrusion Detection Specialists

"Right now, in its current form, I don't believe that the current products are mature enough to be performing active response. The main problem is that signature-based

systems are still very prone to false-positives, and any device that is re-configuring infrastructure equipment (shunning) could easily be turned into a denial of service tool."

Greg Shipley, Network Computing online journal (http://www.nwc.com/)

"As well as alerting to an attack occurring some IDS can defend against them, this is achieved in a variety of ways. Firstly by injecting packets to reset the connection or alternatively by reconfiguring routers and firewalls to reject future traffic from the same address. There are problems with both methods in order to inject packets the IDS needs to have an active interface thereby making itself susceptible to attack. See stealth. There are ways around this, such as having the active interface inside the firewall. As for the latter method of automated response, it isn't unknown for attackers to abuse the latter method by spoofing the address of a friendly party and launching an attack, the IDS then configures the routers/firewalls to reject the these addresses, effectively DOSing themselves. "

Andy Cuff (Talisker), http://www.networkintrusion.co.uk,

"Proactive measures are a reasonable idea, unless they can be subverted. For instance, if you decide to shut down your network connections as a proactive approach, then an intrusion attempt can be used as a denial of service"

Gene Spafford, <spaf@cerias.purdue.edu>

"Various methods of automated response are possible. Everything from just noting the problem, perhaps changing firewall or router rules and configurations, scanning the host that is attacking you, etc.

All of these are areas of research at many institutions. There has not been much published, as far as I can tell."

Tom E. Perrine, San Diego Supercomputer Center

APPENDIX B

APPENDIX B

REVIEW OF CURRENT INTRUSION TAXONOMIES

Previous research has given rise to a number of intrusion taxonomies, each of which presents an alternative view of the situation. Brief summaries of a number of notable approaches are given below.

Cheswick and Bellovin classification of attacks

Cheswick and Bellovin in their text on firewalls (Cheswick and Bellovin 1994) have classified attacks into the seven categories listed in Table B.1.

1. **Stealing passwords** - methods used to obtain other users' passwords
2. **Social engineering** - talking your way into information that you should not have
3. **Bugs and backdoors** - taking advantage of systems that do not meet their specifications, or replacing software with compromised versions
4. **Authentication failures** - defeating of mechanisms used for authentication
5. **Protocol failures** - protocols themselves are improperly designed or implemented
6. **Information leakage** - using systems such as finger or the DNS to obtain information that is necessary to administrators and the proper operation of the network, but could also be used by attackers
7. **Denial-of-service** - efforts to prevent users from being able to use their systems.

Table B.1 Seven Categories of attacks

Although this approach provides a general overview, including the main categories of intrusions, it seems too general and does not give any insight into the relationship between different classes of attacks or their different characteristics.

Result-based Taxonomies

Another approach that considers the characteristics of attacks is the conception of result-based taxonomies. In such approach, all attacks are grouped into basic categories according to their result, aiming to give more insight into the severity of attacks. An example is a taxonomy devised by Cohen (1995) that includes categories such as:

- **Corruption** - unauthorised modification of information;
- **Leakage** - when information ends up where it should not be;
- **Denial** - when computer or network services are not available for use.

Another example is the taxonomy of Russell and Gangemi (1991), who define the categories by using opposite terms. That is:

- **Secrecy and confidentiality** instead of leakage;
- **Accuracy, integrity, and authenticity** instead of corruption;
- **Availability** instead of denial.

Although result-based taxonomies can be useful, as the result of an attack is important in the response process, they are too general to provide a meaningful association between different types of attacks and their impact. Whether an intrusion leads to corruption, leakage, or denial this is not the only characteristic relevant to intrusion response. It is necessary to have some classification that will provide a more detailed picture of the incidents and their results.

SRI Neumann-Parker Taxonomy

Neumann and Parker (1989) developed an intrusion taxonomy based upon a large number of incidents (about 3,000) reported to the Internet Risks Forum. The taxonomy classifies intrusions into nine categories, according to key elements that might indicate a particular type of incident. Table B.2 summarises the overall scheme.

NP1 External Misuse	Generally non-technological and unobserved, physically separate from computer and communication facilities, for example visual spying.
NP2 Hardware Misuse	a) Passive, with no (immediate) side effects b) Active, with side effects.
NP3 Masquerading	Impersonation; playback and spoofing attacks etc.
NP4 Setting up Subsequent Misuse	Planting and arming malicious software.
NP5 Bypassing Intended Controls	Circumvention of existing controls or improper acquisition of otherwise denied authority.
NP6 Active Misuse of Resources	Misuse of (apparently) conferred authority that alters the system or its data.
NP7 Passive Misuse of Resources	Misuse of (apparently) conferred reading authority.
NP8 Misuse Resulting from Inaction	Failure to avert a potential problem in a timely fashion, or an error of omission, for example.
NP9 Use as an Indirect Aid in Committing other Misuse	a) As a tool in planning computer misuse etc. b) As a tool in planning criminal/unethical activity.

Source: (Neumann and Parker 1989)

Table B.2 SRI Neumann-Parker taxonomy

An extension of the Neumann-Parker taxonomy was produced by Lindqvist and Jonsson (1997), which further refines security incidents into intrusions, attacks and breaches. It examines these issues from a system-owner perspective, based upon a number of laboratory experiments (60 separate cases of breaches). The results of these experiments indicated a need for further subdivision of the Neumann-Parker classes 5, 6 and 7, as shown in Table B.3 below. Their work provides further insight into attack techniques, and therefore, aids the process of spotting aspects of system and network activity that might indicate an intrusion.

NP5 Bypassing Intended Controls	Password attacks	Capture
		Guessing
	Spoofing privileged programs,	
	Utilizing weak authentication	
NP6 Active Misuse of Resources	Exploiting inadvertent write permission	
	Resource exhaustion	
NP7 Passive Misuse of Resources	Manual browsing	
	Automated browsing	Using a personal tool
		Using a publicly available tool

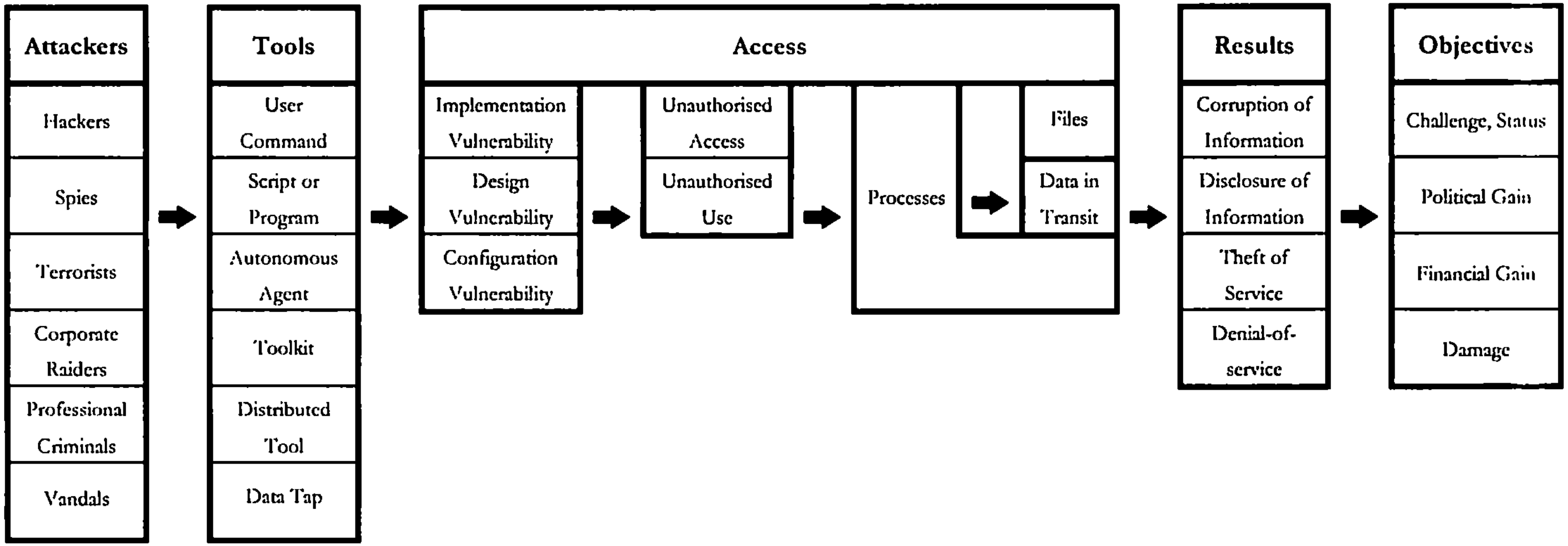
Source: (Lindqvist and Jonsson 1997)

Table B.3 Lindqvist and Jonsson extension of the Neumann-Parker taxonomy

The main contribution of the Neumann - Parker, Lindqvist - Jonsson taxonomies is the fact that they are both based upon a considerable number of real incidents, and hence, can provide a useful insight into the different types of intrusions. Also, they provide useful insight into the techniques of an attack, and could prove useful in an intrusion detection context. However, in the context of intrusion response, it would be desirable to look into more intrusion characteristics, if meaningful associations are to be made.

Howard taxonomy of computer and network attacks

Howard (1997) follows a different approach by focusing upon the process of an attack, rather than classification categories. Howard's taxonomy establishes a link through the different potential *attackers* (classified as hackers, spies, terrorists, corporate raiders, professional criminals and vandals) and the *tools* and *access* methods that they may utilise, leading to the *results* that enable the attackers to achieve their *objectives*. The details of the taxonomy are presented in Figure B.1. This taxonomy was also based upon the analysis of real incidents, as reported to the CERT/CC, from 1989 to 1995 (7649 incidents, according to CERT statistics). Thus it also represents a valuable tool for studying attacks and their characteristics. However, even if it provides a greater level of detail about contextual factors regarding intrusions, it does not present a comprehensive top-level classification of intrusion incidents, or yield an appropriate classification that could be used to determine the required response – a criticism that could also be levelled at the other examples considered here.



Source: (Howard 1997)

Figure B.1 Process-Based Attack Taxonomy

APPENDIX C

APPENDIX C

RESPONSE POLICY RULES

```
;;;======================================================
;;;   Response Expert Sample Problem
;;;
;;;     FAIR: Flexible Automated Intelligent Responder.
;;;     This example selects appropriate responses
;;;     based on the Response Policy.
;;;
;;;     CLIPS Version 6.0 Example
;;;
;;;     To execute, load, reset and run.
;;;======================================================

(defmodule MAIN (export ?ALL))

;;****************
;;* DEFGLOBALS   *
;;****************
(defglobal
      ?*theQuestion* = "")
(defglobal
      ?*theAnswer* = "")
(defglobal
      ?*allowedValues* = "")
(defglobal
     ?*theAttribute* = "")
(defglobal
      ?*needInput* = 0)

;;*****************
;;* INITIAL STATE *
;;*****************

(deftemplate MAIN::attribute
   (slot name)
   (slot value)
   (slot certainty (default 100.0)))

(defrule MAIN::start
  (declare (salience 10000))
  =>
  (set-fact-duplication TRUE)
  (focus QUESTIONS CHOOSE-QUALITIES RESPONSES PRINT-RESULTS))

(defrule MAIN::combine-certainties ""
  (declare (salience 100)
           (auto-focus TRUE))
  ?rem1 <- (attribute (name ?rel&~best-name) (value ?val) (certainty ?per1))
  ?rem2 <- (attribute (name ?rel&~best-name) (value ?val) (certainty ?per2))
  (test (neq ?rem1 ?rem2))
  =>
  (retract ?rem1)
  (modify ?rem2 (certainty (/ (- (* 100 (+ ?per1 ?per2)) (* ?per1 ?per2)) 100))))

(defrule MAIN::combine-name-certainties ""
  (declare (salience 100)
           (auto-focus TRUE))
  ?rem1 <- (attribute (name best-name) (value ?val) (certainty ?per1))
  ?rem2 <- (attribute (name best-name) (value ?val) (certainty ?per2))
  (test (neq ?rem1 ?rem2))
  =>
  (retract ?rem1)
  (modify ?rem2 (certainty (+ ?per1 ?per2))))


;;*****************
;;* QUESTION RULES *
;;*****************

(defmodule QUESTIONS (import MAIN ?ALL) (export ?ALL))

(deftemplate QUESTIONS::question
   (slot attribute (default ?NONE))
   (slot the-question (default ?NONE))
   (multislot valid-answers (default ?NONE))
   (slot already-asked (default FALSE))
   (multislot precursors (default ?DERIVE)))

(defrule QUESTIONS::ask-a-question
   ?f <- (question (already-asked FALSE)
                   (precursors)
                   (the-question ?the-question)
                   (attribute ?the-attribute)
                   (valid-answers $?valid-answers))
   =>
   (modify ?f (already-asked TRUE))
   (bind ?*needInput* 1)
   (bind ?*theQuestion* ?the-question)
   (bind ?*allowedValues* ?valid-answers)
   (bind ?*theAttribute* ?the-attribute)
   (halt))

(defrule QUESTIONS::precursor-is-satisfied
   ?f <- (question (already-asked FALSE)
                   (precursors ?name is ?value $?rest))
         (attribute (name ?name) (value ?value))
   =>
   (if (eq (nth 1 ?rest) and)
    then (modify ?f (precursors (rest$ ?rest)))
    else (modify ?f (precursors ?rest))))

(defrule QUESTIONS::precursor-is-not-satisfied
   ?f <- (question (already-asked FALSE)
                   (precursors ?name is-not ?value $?rest))
         (attribute (name ?name) (value ~?value))
   =>
   (if (eq (nth 1 ?rest) and)
    then (modify ?f (precursors (rest$ ?rest)))
    else (modify ?f (precursors ?rest))))

;;*********************
;;* RESPONSE QUESTIONS *
;;*********************

(defmodule RESPONSE-QUESTIONS (import QUESTIONS ?ALL))

(deffacts RESPONSE-QUESTIONS::question-attributes
  (question (attribute alarm)
            (the-question "alarm?")
            (valid-answers successful-user-login virus))
  (question (attribute alarm-confidence)
            (the-question "alarm-confidence?")
            (valid-answers low medium high))
  (question (attribute alert-status)
            (the-question "alert-status?")
            (valid-answers low medium high))
  (question (attribute overall-threat)
            (the-question "overall-threat?")
            (valid-answers low medium high))
  (question (attribute urgency)
            (the-question "urgency?")
            (valid-answers low medium high))
  (question (attribute responder-efficiency)
            (the-question "responder-efficiency?")
            (valid-answers low medium high))
  (question (attribute detection-efficiency)
            (the-question "detection-efficiency?")
            (valid-answers low medium high))
  (question (attribute has-perpetrator)
            (the-question "has-perpetrator?")
            (valid-answers yes no))
  (question (attribute perpetrator-threat)
            (precursors has-perpetrator is yes)
            (the-question "perpetrator-threat?")
            (valid-answers low medium high))
  (question (attribute insider-perpetrator)
            (precursors has-perpetrator is yes)
            (the-question "insider-perpetrator?")
            (valid-answers yes no))
  (question (attribute target-importance)
            (the-question "target-importance?")
            (valid-answers low medium high))
  (question (attribute target-dependant-systems)
            (the-question "target-dependant-systems?")
            (valid-answers low medium high))
  (question (attribute number-systems-at-risk)
            (the-question "number-systems-at-risk?")
            (valid-answers low medium high))
  (question (attribute target-usage)
            (the-question "target-usage?")
            (valid-answers low medium high))
  (question (attribute has-user-account)
            (the-question "has-user-account?")
            (valid-answers yes no))
  (question (attribute privileged-user-account)
            (precursors has-user-account is yes)
            (the-question "privileged-user-account?")
            (valid-answers yes no))
  (question (attribute target-vulnerable)
            (the-question "target-vulnerable?")
            (valid-answers yes no))
  (question (attribute target-has-critical-information)
            (the-question "target-has-critical-information?")
            (valid-answers yes no))
  (question (attribute target-offer-critical-operation)
            (the-question "target-offer-critical-operation?")
            (valid-answers yes no))
  (question (attribute target-role)
            (the-question "target-role?")
            (valid-answers internal-server external-server user-workstation
network-component))
  (question (attribute target-idle)
            (the-question "target-idle?")
            (valid-answers yes no))
  (question (attribute critical-applications-running)
            (the-question "critical-applications-running?")
            (valid-answers yes no))
  (question (attribute critical-files-open)
            (the-question "critical-files-open?")
            (valid-answers yes no))
  (question (attribute auditing-sw-running)
            (the-question "auditing-sw-running?")
            (valid-answers yes no))
  (question (attribute vulnerable-sw-running)
            (the-question "vulnerable-sw-running?")
            (valid-answers yes no))
  (question (attribute intrusion-confidence)
            (the-question "intrusion-confidence?")
            (valid-answers low medium high))

  (question (attribute abnormal-time)
            (the-question "abnormal-time?")
            (valid-answers yes no))
)

;;******************
;; The RULES module
;;******************

(defmodule RULES (import MAIN ?ALL) (export ?ALL))

(deftemplate RULES::rule
  (slot certainty (default 100.0))
  (multislot if)
  (multislot then))

(defrule RULES::throw-away-ands-in-antecedent
  ?f <- (rule (if and $?rest))
  =>
  (modify ?f (if ?rest)))

(defrule RULES::throw-away-ands-in-consequent
  ?f <- (rule (then and $?rest))
  =>
  (modify ?f (then ?rest)))

(defrule RULES::remove-is-condition-when-satisfied
  ?f <- (rule (certainty ?c1)
              (if ?attribute is ?value $?rest))
  (attribute (name ?attribute)
             (value ?value)
             (certainty ?c2))
  =>
  (modify ?f (certainty (min ?c1 ?c2)) (if ?rest)))

(defrule RULES::remove-is-not-condition-when-satisfied
  ?f <- (rule (certainty ?c1)
              (if ?attribute is-not ?value $?rest))
  (attribute (name ?attribute) (value ~?value) (certainty ?c2))
  =>
  (modify ?f (certainty (min ?c1 ?c2)) (if ?rest)))

(defrule RULES::perform-rule-consequent-with-certainty
  ?f <- (rule (certainty ?c1)
              (if)
              (then ?attribute is ?value with certainty ?c2 $?rest))
  =>
  (modify ?f (then ?rest))
  (assert (attribute (name ?attribute)
                     (value ?value)
                     (certainty (/ (* ?c1 ?c2) 100)))))

(defrule RULES::perform-rule-consequent-without-certainty
  ?f <- (rule (certainty ?c1)
              (if)
              (then ?attribute is ?value $?rest))
  (test (or (eq (length$ ?rest) 0)
            (neq (nth 1 ?rest) with)))
  =>
  (modify ?f (then ?rest))
  (assert (attribute (name ?attribute) (value ?value) (certainty ?c1))))


;;**********************************
;;* CHOOSE RESPONSE ATTRIBUTES RULES *
;;**********************************

(defmodule CHOOSE-QUALITIES (import RULES ?ALL)
                            (import QUESTIONS ?ALL)
                            (import MAIN ?ALL))

(defrule CHOOSE-QUALITIES::startit => (focus RULES))

(deffacts CHOOSE-QUALITIES::the-response-rules

;; Other rules - need-attributes=1: Efficiency is more important
              ;- need-attributes=2: Efficiency should be balanced with
              ;  transparency and countereffects
              ;- need-attributes-3: Highest transparency and
              ;  lowest countereffects are more important

(rule (if urgency is high)
      (then need-attributes is 1 with certainty 40))

(rule (if urgency is medium)
      (then need-attributes is 2 with certainty 40))

(rule (if urgency is low)
      (then need-attributes is 3 with certainty 40))

(rule (if overall-threat is high)
      (then need-attributes is 1 with certainty 40))

(rule (if overall-threat is medium)
      (then need-attributes is 2 with certainty 40))

(rule (if overall-threat is low)
      (then need-attributes is 3 with certainty 40))

;; Rules for Best Phase
;  forestall, collectEvidence whenever a forestall, collectEvidence option is
;  available, use it
(rule (if)
      (then best-phase is forestall with certainty 100 and
            best-phase is collectEvidence with certainty 100))

    ; Notify
(rule (if responder-efficiency is low and alert-status is high)
      (then best-phase is notification with certainty 60))

(rule (if responder-efficiency is low and alert-status is medium)
      (then best-phase is notification with certainty 50))

(rule (if responder-efficiency is low and urgency is high)
      (then best-phase is notification with certainty 40))

(rule (if responder-efficiency is low and urgency is medium)
      (then best-phase is notification with certainty 30))

(rule (if responder-efficiency is low and overall-threat is high)
      (then best-phase is notification with certainty 60))

(rule (if responder-efficiency is low and overall-threat is medium)
      (then best-phase is notification with certainty 50))

(rule (if responder-efficiency is low and target-importance is high)
     (then best-phase is notification with certainty 40))

(rule (if responder-efficiency is low and target-importance is medium)
     (then best-phase is notification with certainty 30))

(rule (if responder-efficiency is medium and alert-status is high)
     (then best-phase is notification with certainty 50))

(rule (if responder-efficiency is medium and urgency is high)
     (then best-phase is notification with certainty 30))

(rule (if responder-efficiency is medium and overall-threat is high)
     (then best-phase is notification with certainty 50))

(rule (if responder-efficiency is medium and target-importance is high)
     (then best-phase is notification with certainty 30))

   ; Investigate, forestall
(rule (if alarm-confidence is-not high and urgency is-not high)
     (then best-phase is investigate with certainty 100 and
           best-phase is collectEvidence with certainty 100))

   ; Protect, Recover, CollectEvidence, Forestall
(rule (if alarm-confidence is high and responder-efficiency is high)
     (then best-phase is protectResources with certainty 100 and
           best-phase is recover with certainty 100 and
           best-phase is collectEvidence with certainty 100))

(rule (if alarm-confidence is high and responder-efficiency is medium)
     (then best-phase is protectResources with certainty 60 and
           best-phase is recover with certainty 60 and
           best-phase is collectEvidence with certainty 60))

(rule (if alarm-confidence is high and responder-efficiency is-not high and
alert-status is high)
     (then best-phase is protectResources with certainty 55 and
           best-phase is recover with certainty 55 and
           best-phase is collectEvidence with certainty 55))

(rule (if alarm-confidence is high and responder-efficiency is-not high and
alert-status is medium)
     (then best-phase is protectResources with certainty 50 and
           best-phase is recover with certainty 50 and
           best-phase is collectEvidence with certainty 50))

(rule (if alarm-confidence is high and responder-efficiency is-not high and
urgency is high)
     (then best-phase is protectResources with certainty 55 and
           best-phase is recover with certainty 55 and
           best-phase is collectEvidence with certainty 55))

(rule (if alarm-confidence is high and responder-efficiency is-not high and
urgency is medium)
     (then best-phase is protectResources with certainty 50 and
           best-phase is recover with certainty 50 and
           best-phase is collectEvidence with certainty 50))

(rule (if alarm-confidence is high and responder-efficiency is-not high and
overall-threat is high)
     (then best-phase is protectResources with certainty 55 and
           best-phase is recover with certainty 55 and
           best-phase is collectEvidence with certainty 55))

(rule (if alarm-confidence is high and responder-efficiency is-not high and
overall-threat is medium)
     (then best-phase is protectResources with certainty 50 and
           best-phase is recover with certainty 50 and
```

```
            best-phase is collectEvidence with certainty 50))

(rule (if alarm-confidence is high and responder-efficiency is-not high and
target-importance is high)
      (then best-phase is protectResources with certainty 55 and
            best-phase is recover with certainty 55 and
            best-phase is collectEvidence with certainty 55))

(rule (if alarm-confidence is high and responder-efficiency is-not high and
target-importance is medium)
      (then best-phase is protectResources with certainty 50 and
            best-phase is recover with certainty 50 and
            best-phase is collectEvidence with certainty 50))

    ;; Rules for maximum Stopping Power

(rule (if urgency is low)
      (then max-stoppingPower is 3 with certainty 20))

(rule (if urgency is medium)
      (then max-stoppingPower is 6 with certainty 20))

(rule (if urgency is high)
      (then max-stoppingPower is 10 with certainty 20))

(rule (if overall-threat is low)
      (then max-stoppingPower is 3 with certainty 20))

(rule (if overall-threat is medium)
      (then max-stoppingPower is 6 with certainty 20))

(rule (if overall-threat is high)
      (then max-stoppingPower is 10 with certainty 20))

(rule (if responder-efficiency is low)
      (then max-stoppingPower is 3 with certainty 50))

(rule (if responder-efficiency is medium)
      (then max-stoppingPower is 6 with certainty 50))

(rule (if responder-efficiency is high)
      (then max-stoppingPower is 10 with certainty 50))

(rule (if alert-status is low)
      (then max-stoppingPower is 3 with certainty 50))

(rule (if alert-status is medium)
      (then max-stoppingPower is 6 with certainty 50))

(rule (if alert-status is high)
      (then max-stoppingPower is 10 with certainty 50))

(rule (if target-importance is low)
      (then max-stoppingPower is 3 with certainty 20))

(rule (if target-importance is medium)
      (then max-stoppingPower is 6 with certainty 20))

(rule (if target-importance is high)
      (then max-stoppingPower is 10 with certainty 20))


    ;; Rules for best name
(rule (if alarm is probe and number-systems-at-risk is low)
      (then  best-name is check-patches-update with certainty 50))

(rule (if alarm is probe and target-vulnerable is no)
      (then  best-name is check-patches-update with certainty 50))

(rule (if alarm is probe and alert-status is high)
      (then  best-name is check-patches-update with certainty 33))

(rule (if alarm is probe and number-systems-at-risk is medium)
      (then  best-name is check-patches-update with certainty 33))

(rule (if alarm is probe and target-vulnerable is no)
      (then  best-name is check-patches-update with certainty 33))

(rule (if alarm is probe and urgency is high)
      (then  best-name is check-patches-update with certainty 33))

(rule (if alarm is probe and number-systems-at-risk is medium)
      (then  best-name is check-patches-update with certainty 33))

(rule (if alarm is probe and target-vulnerable is no)
      (then  best-name is check-patches-update with certainty 33))

(rule (if alarm is probe and alert-status is medium)
      (then  best-name is alert with certainty 33))

(rule (if alarm is probe and number-systems-at-risk is medium)
      (then  best-name is alert with certainty 33))

(rule (if alarm is probe and target-vulnerable is no)
      (then  best-name is alert with certainty 33))

(rule (if alarm is probe and urgency is medium)
      (then  best-name is alert with certainty 33))

(rule (if alarm is probe and number-systems-at-risk is medium)
      (then  best-name is alert with certainty 33))

(rule (if alarm is probe and target-vulnerable is no)
      (then  best-name is alert with certainty 33))

(rule (if alarm is successful-user-login and target-importance is high)
      (then  best-name is fingerprint-recognition with certainty 50 and
             best-name is disconnect-user with certainty 50))

(rule (if alarm is successful-user-login and privileged-user-account is yes)
      (then  best-name is fingerprint-recognition with certainty 50 and
             best-name is disconnect-user with certainty 50))

(rule (if alarm is successful-user-login and target-offer-critical-operation is
yes)
      (then  best-name is back-up-unsaved-work with certainty 50 and
             best-name is limit-critical-applications-access with certainty 50))

(rule (if alarm is successful-user-login and privileged-user-account is yes)
      (then  best-name is back-up-unsaved-work with certainty 50 and
             best-name is limit-critical-applications-access with certainty 50))

(rule (if alarm is successful-user-login and privileged-user-account is yes)
      (then  best-name is facial-recognition with certainty 100 and
             best-name is continuous-keystroke-analysis with certainty 100 and
             best-name is cognitive-questions-authentication with certainty 100
and
             best-name is alert with certainty 100 and
             best-name is redirect-to-decoy-system with certainty 100 and
             best-name is back-up-unsaved-work with certainty 100 and
             best-name is disconnect-user with certainty 100))

(rule (if alarm is successful-user-login and critical-files-open is yes)
      (then  best-name is back-up-critical-information with certainty 100))
```

```
(rule (if alarm is successful-user-login and target-importance is-not low)
      (then  best-name is continuous-keystroke-analysis with certainty 100 and
             best-name is cognitive-questions-authentication with certainty 100))

(rule (if alarm is successful-user-login and target-offer-critical-operation is
yes)
      (then  best-name is back-up-unsaved-work with certainty 100))

(rule (if alarm is successful-user-login and target-idle is yes)
      (then  best-name is disconnect-user with certainty 100))

(rule (if alarm is successful-user-login and target-role is network-component)
      (then  best-name is disconnect-user with certainty 100))

(rule (if alarm is successful-user-login and critical-applications-running is
yes)
      (then  best-name is back-up-unsaved-work with certainty 100 and
             best-name is limit-critical-applications-access with certainty 100))

(rule (if alarm is successful-user-login)
      (then  best-name is keystroke-analysis with certainty 100 and
             best-name is log-session with certainty 100 and
             best-name is add-entry-to-log-file with certainty 100))

)


;;****************************
;;* RESPONSE SELECTION RULES *
;;****************************

(defmodule RESPONSES (import MAIN ?ALL))

(deffacts any-attributes
  (attribute (name best-name) (value any))
  (attribute (name best-phase) (value any) (certainty 0))
  (attribute (name max-stoppingPower) (value 0) (certainty 0))
  (attribute (name need-attributes) (value 2) (certainty 50)))

(deftemplate RESPONSES::response
  (slot name (default ?NONE))
  (slot desc (default any))
  (slot phase (default any))
  (slot ceffects (default 0))
  (slot stoppingPower (default 0))
  (slot transparency (default 0))
  (slot efficiency (default 0))
  (slot already-asserted (default FALSE)))

(deffacts RESPONSES::the-response-list
  (response (name resa) (desc add-entry-to-log-file) (phase forestall) (ceffects
1) (stoppingPower 0) (transparency 9) (efficiency 80))
  (response (name resb) (desc redirect-to-decoy-system) (phase protectResources)
(ceffects 18) (stoppingPower 7) (transparency 5) (efficiency 50))
  (response (name resc) (desc check-patches-update) (phase forestall) (ceffects
5) (stoppingPower 3) (transparency 8) (efficiency 80))
  (response (name resd) (desc keystroke-analysis) (phase investigate) (ceffects
6) (stoppingPower 3) (transparency 7) (efficiency 60))
  (response (name rese) (desc facial-recognition) (phase investigate) (ceffects
4) (stoppingPower 3) (transparency 8) (efficiency 50))
  (response (name resf) (desc continuous-keystroke-analysis) (phase investigate)
(ceffects 16) (stoppingPower 3) (transparency 7) (efficiency 80))
  (response (name resg) (desc cognitive-questions-authentication) (phase
investigate) (ceffects 7) (stoppingPower 3) (transparency 1) (efficiency 70))
  (response (name resh) (desc fingerprint-recognition) (phase investigate)
(ceffects 8) (stoppingPower 3) (transparency 0) (efficiency 90))
```

```
  (response (name resi) (desc log-session) (phase collectEvidence) (ceffects 4)
(stoppingPower 2) (transparency 9) (efficiency 90))
  (response (name resj) (desc alert) (phase notification) (ceffects 3)
(stoppingPower 0) (transparency 9) (efficiency 90))
  (response (name resk) (desc redirect-to-decoy-system) (phase protectResources)
(ceffects 18) (stoppingPower 7) (transparency 5) (efficiency 50))
  (response (name resl) (desc patch-vulnerable-systems) (phase forestall)
(ceffects 7) (stoppingPower 8) (transparency 4) (efficiency 90))
  (response (name resm) (desc patch-vulnerable-systems-background) (phase
forestall) (ceffects 6) (stoppingPower 8) (transparency 9) (efficiency 90))
  (response (name resn) (desc reset-connection) (phase protectResources)
(ceffects 12) (stoppingPower 8) (transparency 2) (efficiency 90))
  (response (name reso) (desc block-system) (phase protectResources) (ceffects
12) (stoppingPower 9) (transparency 1) (efficiency 95))
  (response (name resp) (desc disconnect-user) (phase protectResources) (ceffects
13) (stoppingPower 9) (transparency 1) (efficiency 95))
  (response (name resq) (desc limit-user-access) (phase forestall) (ceffects 5)
(stoppingPower 5) (transparency 8) (efficiency 80))
  (response (name resr) (desc back-up-unsaved-work) (phase forestall) (ceffects
3) (stoppingPower 1) (transparency 8) (efficiency 90))
  (response (name ress) (desc back-up-critical-information) (phase forestall)
(ceffects 3) (stoppingPower 1) (transparency 8) (efficiency 90))
  (response (name rest) (desc limit-critical-applications-access) (phase
forestall) (ceffects 5) (stoppingPower 5) (transparency 8) (efficiency 80))
)


(defrule RESPONSES::generate-responses-1
(declare (salience 97))
  (attribute (name best-name) (value ?b) (certainty ?certainty-1))
  (attribute (name best-phase) (value ?p) (certainty ?certainty-2))
  (attribute (name max-stoppingPower) (value ?s) (certainty ?certainty-3))
  (attribute (name need-attributes) (value 1))
  ?rem <- (response (name ?name)
             (desc ?b)
            (phase ?p)
             (efficiency ?e)
             (stoppingPower ?sr&:(<= ?sr ?s))
             (already-asserted FALSE))
  (response (name ?name1&:(neq ?name ?name1))
             (desc ?b1)
            (phase ?p)
             (efficiency ?e1)
             (stoppingPower ?sr&:(<= ?sr ?s)))
  ?rem1 <- (attribute (name candidate-response) (value ?b))
  =>
  (retract ?rem1)
  (modify ?rem (already-asserted TRUE))
  (assert (attribute (name response) (value ?b)
                     (certainty (* (min ?certainty-2 ?certainty-3) (/ ?e
100))))))


(defrule RESPONSES::generate-responses-2
(declare (salience 97))
  (attribute (name best-name) (value ?b) (certainty ?certainty-1))
  (attribute (name best-phase) (value ?p) (certainty ?certainty-2))
  (attribute (name max-stoppingPower) (value ?s) (certainty ?certainty-3))
  (attribute (name need-attributes) (value 2))
  ?rem <- (response (name ?name)
             (desc ?b)
            (phase ?p)
             (ceffects ?c) (transparency ?t) (efficiency ?e)
             (stoppingPower ?sr&:(<= ?sr ?s))
             (already-asserted FALSE))
  (response (name ?name1&:(neq ?name ?name1))
             (desc ?b1)
            (phase ?p)
             (ceffects ?c1) (transparency ?t1) (efficiency ?e1)
             (stoppingPower ?sr&:(<= ?sr ?s)))
  ?rem1 <- (attribute (name candidate-response) (value ?b))
  =>
  (retract ?rem1)
  (modify ?rem (already-asserted TRUE))
  (assert (attribute (name response) (value ?b)
                     (certainty (* (min ?certainty-2 ?certainty-3) (/ (+ (* (+ (-
30 ?c) ?t) 2.5) ?e) 200))))))


(defrule RESPONSES::generate-responses-3
(declare (salience 97))
  (attribute (name best-name) (value ?b) (certainty ?certainty-1))
  (attribute (name best-phase) (value ?p) (certainty ?certainty-2))
  (attribute (name max-stoppingPower) (value ?s) (certainty ?certainty-3))
  (attribute (name need-attributes) (value 3))
  ?rem <- (response (name ?name)
             (desc ?b)
            (phase ?p)
             (ceffects ?c) (transparency ?t)
             (stoppingPower ?sr&:(<= ?sr ?s))
             (already-asserted FALSE))
  (response (name ?name1&:(neq ?name ?name1))
             (desc ?b1)
            (phase ?p)
             (ceffects ?c1) (transparency ?t1)
             (stoppingPower ?sr&:(<= ?sr ?s)))
  ?rem1 <- (attribute (name candidate-response) (value ?b))
  =>
  (retract ?rem1)
  (modify ?rem (already-asserted TRUE))
  (assert (attribute (name response) (value ?b)
                     (certainty (* (min ?certainty-2 ?certainty-3) (/ (+ (- 30
?c) ?t) 40))))))

(defrule RESPONSES::generate-responses
(declare (salience 97))
  (attribute (name best-name) (value ?b) (certainty ?certainty-1))
  (attribute (name best-phase) (value ?p) (certainty ?certainty-2))
  (attribute (name max-stoppingPower) (value ?s) (certainty ?certainty-3))
  (response (name ?name)
             (desc ?b)
            (phase ?p)
             (ceffects ?c) (transparency ?t)
             (stoppingPower ?sr&:(<= ?sr ?s)))
  ?rem <- (attribute (name candidate-response) (value ?b))
  (not (response (name ?name1&:(neq ?name ?name1))
             (desc ?b1)
            (phase ?p)
             (ceffects ?c1) (transparency ?t1)
             (stoppingPower ?sr&:(<= ?sr ?s))))
  =>
  (assert (attribute (name response) (value ?b)
                     (certainty (min ?certainty-2 ?certainty-3))))
   (retract ?rem))

(defrule RESPONSES::select-response-names
(declare (salience 100))
  ?rem <- (attribute (name best-name) (value ?val1) (certainty ?per1&:(< ?per1
95)))
  =>
  (retract ?rem))

(defrule RESPONSES::add-candidate-responses
(declare (salience 99))
  (attribute (name best-name) (value ?b) (certainty ?certainty-1))
  (response (name ?name)
            (desc ?b)
            (phase ?p)
            (transparency ?t)
            (stoppingPower ?sr)
            (already-asserted FALSE))
  =>
  (assert (attribute (name candidate-response) (value ?b)
                     (certainty (+ ?t ?sr)))))

(defrule RESPONSES::select-max-stoppingPower
(declare (salience 98))
  ?rem <- (attribute (name max-stoppingPower) (value ?val) (certainty ?per))
  ?rem1 <- (attribute (name max-stoppingPower) (value ?val1) (certainty ?per1&:(>
?per ?per1)))
  =>
  (retract ?rem1))

(defrule RESPONSES::select-appropriate-attributes
(declare (salience 98))
  ?rem <- (attribute (name need-attributes) (value ?val) (certainty ?per))
  ?rem1 <- (attribute (name need-attributes) (value ?val1) (certainty ?per1&:(>
?per ?per1)))
  =>
  (retract ?rem1))


(defrule RESPONSES::remove-weak-phases
(declare (salience 98))
  ?rem <- (attribute (name best-phase) (value ?val) (certainty ?per&:(< ?per
50)))
  =>
  (retract ?rem))


(defrule RESPONSES::remove-additional-investigation-choices
(declare (salience 96))
  (attribute (name response) (value ?b) (certainty ?per))
  (response (desc ?b)
            (phase investigate))
  ?rem1 <- (attribute (name response) (value ?b1) (certainty ?per1&:(> ?per
?per1)))
  (response (desc ?b1)
            (phase investigate))
  =>
  (assert (attribute (name candidate-response) (value ?b1) (certainty ?per1)))
  (retract ?rem1))


(defrule RESPONSES::remove-poor-response-choices
(declare (salience 96))
  ?rem <- (attribute (name response) (value ?b) (certainty ?per&:(< ?per 50)))
  =>
  (assert (attribute (name candidate-response) (value ?b) (certainty ?per)))
  (retract ?rem))


;;********************************
;;* PRINT SELECTED RESPONSE RULES *
;;********************************

(defmodule PRINT-RESULTS (import MAIN ?ALL))

(defrule PRINT-RESULTS::header ""
   (declare (salience 10))
   =>
   (printout wdialog crlf)
   (printout wdialog "        ISSUED RESPONSES" crlf)
   (assert (phase print-responses)))

(defrule PRINT-RESULTS::print-response ""
   (declare (salience 9))
  ?rem <- (attribute (name response) (value ?name) (certainty ?per))
  (not (attribute (name response) (certainty ?perl&:(> ?per1 ?per))))
  =>
  (retract ?rem)
  (format wdialog " %-24s %-2d%%%n" ?name ?per)
  (printout wdialog crlf))

(defrule PRINT-RESULTS::header1 ""
   (declare (salience 8))
   =>
   (printout wdialog crlf)
   (printout wdialog "        CANDIDATE RESPONSES" crlf)
   (assert (phase print-responses)))

(defrule PRINT-RESULTS::print-candidate-response ""
   (declare (salience 7))
  ?rem <- (attribute (name candidate-response) (value ?name) (certainty ?per))

  (not (attribute (name candidate-response) (certainty ?perl&:(> ?per1 ?per))))
  =>
  (retract ?rem)
  (format wdialog " %-24s %-2d%%%n" ?name ?per)
  (printout wdialog crlf))

(defrule PRINT-RESULTS::end-spaces ""
   (not (attribute (name response)))
   =>
   (printout wdialog crlf))
```

LIST OF ABBREVIATIONS

LIST OF ABBREVIATIONS

AAIRS	Adaptive, Agent-based, Intrusion Response System
ARB	Automated Response Broker
CERT/CC	CERT Coordination Centre at Carnegie Mellon University
CLIPS	C Language Integrated Production System
CSI	Computer Security Institute
CVE	Common Vulnerabilities and Exposures. A list of standardized names for vulnerabilities and other information security exposures.
DARPA	Defense Advanced Research Projects Agency
DNS	Domain Name Server
DoS	Denial of Service attack
DTI	Department of Trade and Industry
EMERALD	Event Monitoring Enabling Responses to Anomalous Live Disturbances
FAIR	Flexible Automated Intelligent Responder
FBI	Federal Bureau of Investigation
FTP	File Transfer Protocol
HTTP	Hyper-Text Transfer Protocol
IDS	Intrusion Detection System
IIS	Internet Information Server
IMS	Intrusion Monitoring System
IP	Internet Protocol
IPS	Intrusion Prevention System
ISS	Internet Security Systems

NAS	Number of Affected Systems
NFS	Network File Server
OS	Operating System
PDA	Personal Digital Assistant
RAID	Recent Advances in Intrusion Detection conference
SANS	SysAdmin, Audit, Network, Security institute
SMS	Short Message Service
SNMP	Simple Network Management Protocol
SQL	Structured Query Language
SWT	Sleepy Watermark Tracing
TCP	Transport Control Protocol
UDP	User Datagram Protocol
VPN	Virtual Private Network
WWW	World Wide Web
WWWD	World Wide War Drive conference

LIST OF REFERENCES

LIST OF REFERENCES

1. Aleph1 (1996), “Smashing The Stack For Fun And Profit”, Phrack online journal, vol. 7, issue 49, 8 November 1996.

2. Allen J., Christie A., Fithen W., McHugh J., Pickel J., and Stoner E. (2000), *State of the Practice of Intrusion Detection Technologies*, Carnegie Mellon University, Technical Report CMU/SEI-99-TR-028, January 2000, http://www.sei.cmu.edu/ publications/documents/99.reports/99tr028/99tr028abstract.html

3. Amoroso E. (1999) *Intrusion Detection: An Introduction to Internet Surveillance, Correlation, Traps, Trace Back, and Response*, Second Printing, Intrusion.Net Books, New Jersey, June 1999.

4. Axelsson S. (1999) “The Base-Rate Fallacy and its Implications for the Difficulty of Intrusion Detection”, in Proceedings of the 6th ACM Conference on Computer and Communications Security, Singapore, 1-4 November 1999: pp. 1-7, http://www.ce.chalmers.se/staff/sax/difficulty.pdf

5. Axelsson S. (2000) “Intrusion Detection Systems: A Taxomomy and Survey”, Technical Report No 99-15, Dept. of Computer Engineering, Chalmers University of Technology, Sweden, 14 March 2000, http://www.ce.chalmers.se/research/Security/Publications/pubs/taxonomy.ps

6. Bace R. and Mell P. (2001) *NIST Special Publication on Intrusion Detection Systems*, National Institute of Standards and Technology (NIST), http://csrc.nist.gov/publications/nistpubs/800-31/sp800-31.pdf

7. Balepin I., Maltsev S, Rowe J, and Levitt K. (2003) "Using Specification-Based Intrusion Detection for Automated Response", in proceedings of the 6th International Symposium RAID 2003 (Recent Advances in Intrusion Detection), Pittsburgh, PA, September 8-10, 2003, http://seclab.cs.ucdavis.edu/papers/Balepin-RAID-03.pdf.

8. Batista E. (2002) "Report: Wi-Fi Networks Too Risky", Wired News, 8 October 2002, http://www.wired.com/news/business/0,1367,55556,00.html

9. Borland J. (2003) "Is your company habouring file-swappers?", ZDNET UK, 16 July 2003, http://news.zdnet.co.uk/business/0,39020645,2137639,00.htm

10. Bradbury D. (2003) "VPN adoption booms as user companies discover new remote-access applications", Computer Weekly magazine, 17 June 2003, http://www.computerweekly.com/articles/article.asp?liArticleID=122567&liArticleTypeID=20&liCategoryID=1&liChannelID=7&liFlavourID=1&sSearch=&nPage=1

11. Brunnstein K., Fischer-Hübner S., and Swimmer M. (1990) "Classification of Computer Anomalies", Proceedings of the 13th National Computer Security Conference, Washington D.C., 1-4 October 1990: pp 374-384.

12. Carver C.A. Jr., Hill J.M.D, and. Pooch U.W. (2001) "Limiting Uncertainty in Intrusion Response", 2nd Annual IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop, West Point New York, June 5-6 2001.

13. Carver C.A. Jr., and Pooch U.W. (2000b), "An Intrusion Response Taxonomy and its Role in Automatic Intrusion Response", IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop, West Point, New York, June 6-7 2000.

14. CERT/CC (1999) "CERT Advisory CA-1999-04 Melissa Macro Virus", 31 March 1999, http://www.cert.org/advisories/CA-1999-04.html

15. CERT/CC (2000) "CERT Advisory CA-1996-21 TCP SYN Flooding and IP Spoofing Attacks", 29 November 2000, http://www.cert.org/advisories/CA-1996-21.html

16. CERT/CC (2002) "CERT Advisory CA-2001-19 'Code Red' Worm Exploiting Buffer Overflow In IIS Indexing Service DLL", 17 January 2002, http://www.cert.org/advisories/CA-2001-19.html

17. CERT/CC (2002b) "Spoofed/Forged Email", 04 September 2002, http://www.cert.org/tech_tips/email_spoofing.html

18. CERT/CC (2002c) "Securing an Internet Domain Server", August 2002, http://www.cert.org/archive/pdf/dns.pdf

19. CERT/CC (2003) "CERT/CC Statistics 1988-2003", 15 July 2003, http://www.cert.org/stats/cert_stats.html.

20. CERT/CC (2003b) "CERT/CC Overview Incident and Vulnerability Trends", 15 May 2003, http://www.cert.org/present/cert-overview-trends/

21. CERT/CC (2004) "CERT/CC Incident Notes", 18 February 2004, http://www.cert.org/incident_notes/

22. Cheswick W.R., and Bellovin S.M. (1994) *Firewalls and Internet Security: Repelling the Wily Hacker,* Addison-Wesley Publishing Company, 1994.

23. Cheung S., Levitt K.N. (1997) "Protecting Routing Infrastructures from Denial of Service Using Cooperative Intrusion Detection", Proceedings of the New Security Paradigms Workshop, Langdale, Cumbria UK, September 23 - 26, 1997, http://riss.keris.or.kr:8080/pubs/contents/proceedings/commsec/283699/

24. Cisco (2003) "Cisco Security Agent: Data Sheet", Cisco Systems, http://www.cisco.com/application/pdf/en/us/guest/products/ps5057/c1650/cdccont_0900aecd800ade37.pdf

25. Cisco (2003b) "Cisco Intrusion Protection: Data Sheet", Cisco Systems, http://www.cisco.com/warp/public/cc/pd/sqsw/sqidsz/prodlit/netra_ds.pdf

26. Cisco (2003c) "Cisco Threat Response: Introduction", Cisco http://www.cisco.com/en/US/products/sw/secursw/ps5054/index.html

27. Cisco (2003d) "Cisco Secure Intrusion Detection System Overview", Cisco Systems, http://www.cisco.com/en/US/products/sw/secursw/ps2113/products_user_guide_chapter09186a008007e973.html

28. Cohen F.B. (1995) *Protection and Security on the Information Superhighway*, John Wiley & Sons.

29. Cohen F.B. (1999) "Simulating Cyber Attacks, Defenses, and Consequences", March 1999, http://all.net/journal/ntb/simulate/simulate.html.

30. Cohen F.B. and Koike D. (2002) "Leading Attackers Through Attack Graphs with Deceptions", Sandia National Laboratories, 29 May 2002, http://all.net/journal/deception/Agraph/Agraph.html

31. Davey J. (1991) "The CCTA risk analysis and management methodology (CRAMM)", Current Perspectives in Healthcare Computing: pp. 360 – 365.

32. Debar H. (2003) "Intrusion Detection FAQ: What is behavior-based intrusion detection?", SANS Institute, 12 June 2003, http://www.sans.org/resources/idfaq/behavior_based.php

33. Denning D.E. (1987) "An Intrusion-Detection Model", IEEE Transactions on Software Engineering, Vol. SE-13, No. 2, pp.222-232, February 1987.

34. Dougherty C. and Householder A. (2002) "CERT Incident Note IN-2002-04: Exploitation of Vulnerabilities in Microsoft SQL Server", CERT Coordination Centre, 23 May 2002, http://www.cert.org/incident_notes/IN-2002-04.html

35. Dowland P.S., Furnell S.M., and Papadaki M. (2002) "Keystroke Analysis as a Method of Advanced User Authentication and Response", in Security in the Information Society: Visions and Perspectives, Ghonaimy M.A., et. al. (eds): pp 215-226.

36. Doyle B. (2003) "Intrusion Detection FAQ: Passive Fingerprinting Utilizing the Telnet Protocol Negotiation data", SANS Institute, 12 June 2003, http://www.sans.org/resources/idfaq/fingerp_telnet.php

37. DTI (2002) *Information Security Breaches Survey 2002: Technical Report*, Department of Trade and Industry, April 2002, http://www.dti.gov.uk/industry_files/pdf/sbsreport_2002.pdf

38. Ernst & Young LLP (2003) *Global Information Security Survey 2003*, http://www.ey.com/global/download.nsf/Australia/AABS_TSRS_Global_Information_Security_Survey_2003/$file/GISS_2003_FF0224.pdf

39. Flinders University (2004) "Information Technology Security Manual: Business Risk Analysis", Information Services Division, http://www.flinders.edu.au/isd/html/other/security.manual/riskAnalysis.html

40. Furnell S.M., Dowland P.S, Illingworth H.M, and Reynolds P.L. (2000b) "Authentication and Supervision: A survey of user attitudes", Journal of Computers & Security, vol. 19, no. 6: pp529-539.

41. Furnell S.M. (1995) *Data Security in European Healthcare Information Systems*, PhD Thesis, University of Plymouth, United Kingdom.

42. Furnell S.M. (2001) *CyberCrime: Vandalising the Information Society*, Addison-Wesley.

43. Furnell S.M., Magklaras G.B., Papadaki M., and Dowland P.S. (2001) "A Generic Taxonomy for Intrusion Specification and Response", in Proceedings of Euromedia 2001, Valencia, Spain, 18-20 April 2001: pp 125-131.

44. Gartner Inc (2003) http://www4.gartner.com/RecognizedUser

45. Giarratano J. and Riley G. (1998) *Expert systems : principles and programming*, 3rd Edition, PWS Publishing, Boston.

46. Golomb G. (2003) "RE: Recent Gartner IDS/IPS report", posted in SecurityFocus Focus-IDS mailing list, 19 June 2003, http://archives.neohapsis.com/archives/sf/ids/2003-q2/0324.html

47. Graham R. (2003) "Evolution of IDS" Internet Secure Systems, March 2003, http://www.issadvisor.com/columns/EvolutionOfIDS/evolutionofids_files/frame.htm

48. Homeros (850 BC) *The Iliad*

49. Honeynet Project (2000) "Know your Enemy: A Forensic Analysis", 23 May 2000, http://project.honeynet.org/papers/forensics/

50. Honeynet Project (2000b) "Know Your Enemy: The Tools and Methodologies of the Script Kiddie", 21 July 2000, http://www.honeynet.org/papers/enemy/

51. Honeynet Project (2003) "Know your enemy: Honeynets", 12 November 2003, http://www.honeynet.org/papers/honeynet/index.html

52. Honeynet Project (2004) "Tools for Honeynets", 28 February 2004, http://www.honeynet.org/tools/index.html

53. Howard J.D. (1997) *An Analysis of Security Incidents on the Internet 1989 - 1995*, PhD thesis, Carnegie Melon University, 7 April 1997, http://www.cert.org/nav/reports.html

54. Insecure.org (2004) “Nmap Security Scanner: Introduction”, http://www.insecure.org/nmap/

55. Intellitactics (2003) “The NSM Story: Prevail in the Relentless Battle for Enterprise Security” Intellitactics, http://www.itactics.com/products/ppp_4.html

56. Internet Security Systems (1997) “ISS X-Force Database: Ping of Death”, 01 January 1997, http://xforce.iss.net/xforce/xfdb/95

57. Internet Security Systems (2001) “RealSecure: Signatures Reference Guide, Version 6.5”, December 2001, http://documents.iss.net/literature/RealSecure/RS_Signatures_6.5.pdf

58. Internet Security Systems (2001b) “Resurgence of "Code Red" Worm Derivatives”, Internet Security Systems Security Alert, 6 August 2001, http://xforce.iss.net/alerts/advise90.php

59. Internet Security Systems (2003) “Proventia Dynamic Threat Protection Appliances: Protection Without Complexity”, Internet Security Systems, http://documents.iss.net/whitepapers/Proventia.pdf

60. Internet Security Systems (2003b) “RealSecure Server Sensor: Frequently Asked Questions”, Version 3.2, Internet Security Systems, http://documents.iss.net/literature/RealSecure/rsss_faq.pdf

61. Internet Security Systems (2003c) "RealSecure SiteProtector Security Fusion Module 2.0: Frequently Asked Questions", Version 2.0, Internet Security Systems, March 2003, http://documents.iss.net/literature/SiteProtector/FusionFAQ_v2.0.pdf

62. Internet Security Systems (2004) "AdvICE: Exploits: DoS", http://www.iss.net/security_center/advice/Exploits/DoS/default.htm

63. Internet Software Consortium (2004) http://www.isc.org/

64. Irakleous I., Furnell S.M., Dowland P.S., and Papadaki M. (2002) "An experimental comparison of secret-based user authentication technologies", Journal of Information Management & Computer Security, vol. 10, no. 3: pp 100-108.

65. Ko C.C.W. (1996) *Execution Monitoring of Security-Critical Programs in a Distributed System: A Specification-Based Approach*, Ph.D. Thesis, University of California, Davis, USA, August 1996.

66. Kumar S., Spafford E. H. (1995), 'A Software Architecture to support Misuse Intrusion Detection', Purdue University, Computer Science Dept., Technical Report CSD-TR-95-009, March 1995, ftp://coast.cs.purdue.edu/pub/COAST/papers/gene-spafford/

67. Laing B.W. (2003) "Intrusion Detection FAQ: How do you implement IDS (network based) in a heavily switched environment?", SANS Institute, 12 June 2003, http://www.sans.org/resources/idfaq/switched.php

68. Lee S.Y.J. (2001) *Methods of response to IT system intrusions*, MSc thesis, University of Plymouth, Plymouth, UK, September 2001.

69. Leiner B.M., Cerf V.G. et al. (2000) "A Brief History of the Internet", Version 3.31, 4 August 2000, http://www.isoc.org/internet/history/brief.shtml.

70. Lindqvist U., and Jonsson E. (1997) "How to Systematically Classify Computer Security Intrusions", in Proceedings of the 1997 IEEE Symposium on Security and Privacy, 4-7 May 1997, IEEE Computer Society Press.

71. Liston K. (2003) "Intrusion Detection FAQ: Can you explain traffic analysis and anomaly detection?", SANS Institute, 12 June 2003, http://www.sans.org/resources/idfaq/anomaly_detection.php

72. Lu X., Wang Y., and Jain A. K. (2003) "Combining Classifiers for Face Recognition", in Proceedings of the 2003 IEEE International Conference on Multimedia & Expo (ICME), vol. III, 6-9 July 2003, Baltimore, MD: pp. 13-16.

73. Mandia K. and Prosise C. (2001) "Incident Response: Investigating Computer Crime", Osborne / McGraw-Hill, , California USA.

74. Messmer E. (2002) "'Intrusion prevention' raises hopes, concerns", Network World Fusion, 4 November 2002, http://www.nwfusion.com/news/2002/1104prevention.html

75. Messmer E. (2003a) “Why IPS products haven’t taken off”, Network World Fusion, 12 May 2003, http://napps.nwfusion.com/weblogs/security/002755.html.

76. Messmer E. (2003b) “Security debate rages” Network World Fusion, 10 June 2003, http://www.nwfusion.com/news/2003/1006ids.html

77. Mitchell T.M. (1997) *Machine Learning*, McGraw-Hill, New York, 1997.

78. MIT Lincoln Laboratory (2001) “DARPA Intrusion Detection Evaluation: Publications”, http://www.ll.mit.edu/IST/ideval/pubs/pubs_index.html

79. Moore D., Paxson V., Savage S., Shannon C., Staniford S., and Weaver N. (2003) “SDSC Press Release: Sapphire/Slammer Worm shatters previous speed records for spreading through the Internet, California computer experts report”, San Diego Supercomputer Centre (SDSC), 4 February 2003, http://www.sdsc.edu/Press/03/020403_SAPPHIRE.html

80. Moore D., Paxson V., Savage S., Shannon C., Staniford S., and Weaver N. (2003b) “Inside the Slammer Worm”, IEEE Security & Privacy, July-August 2003, Vol.1 No.4: pp 33-39, http://www.computer.org/security/v1n4/j4wea.htm

81. Mounji A. (1997) 'Rule-based Distributed Intrusion Detection', Institut d'Informatique, University of Namur, Belgium, July 1997.

82. NIPC (2001) “Overview of Scans and DDoS Attacks: Executive Summary”, National Infrastructure Protection Centre, USA, http://www.nipc.gov/ddos.pdf

83. Network Associates (2003a) “McAfee Intrushield Security Management: Data Sheet”, Network Associates, http://www.nai.com/us/_tier2/products/_media/sniffer/ds_intrushieldsecuritymanagement.pdf

84. Network Associates (2003b) “McAfee Entercept 4.0: Data Sheet”, Network Associates, http://www.nai.com/us/_tier2/products/_media/sniffer/ds_entercept40.pdf

85. Network Associates (2003c) “McAfee Entercept Standard Edition: Data Sheet”, Network Associates, http://www.nai.com/us/_tier2/products/_media/sniffer/ds_entercept_standardedition.pdf

86. Network Associates (2003d) “The Path to Prevention: White Paper”, Network Associates, http://www.nai.com/

87. Neumann P.G., and Parker D.B. (1989) “A summary of computer misuse techniques”, in Proceedings of the 12th National Computer Security Conference, Baltimore, USA, 10-13 October 1989: pp. 396-407.

88. Newman D., Snyder J. and Thayer R. (2002) “Crying wolf: False alarms hide attacks”, Network World Fusion, 24 June 2002, http://www.nwfusion.com/techinsider/2002/0624security1.html

89. NFR Security (2003) *NFR Sentivist Version 4: Product Overview* NFR Security, http://www.nfr.com/solutions/NFR-FS-Product.pdf

90. Northcutt S. (1999) *Network Intrusion Detection: An Analyst's Handbook,* New Riders Publishing, 22 July 1999: pp 112-114.

91. Papadaki M., Furnell S.M., Lines B.M., and Reynolds P.L. (2002) "A Response-Oriented Taxonomy of IT System Intrusions", Proceedings of Euromedia 2002, Modena, Italy: pp 87-95.

92. Phreak Accident (1993) "Playing Hide and Seek, Unix Style", Phrack Magazine, Vol.4, Issue 43, File 14 of 27, http://project.honeynet.org/papers/enemy3 /hide-n-seek.html.

93. Phyo, A. H. and Furnell, S.M. (2004) "A conceptual framework for monitoring insider misuse", In the Proceedings of Euromedia 2004, Huize Corswarem, Hasselt, Belgium, 21-23 April 2004:pp 90-95.

94. Porras P.A. and Neumann P.G. (1997) 'EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances', in Proceedings of the 20th National Information Systems Security Conference, 9 October 1997, http://www.sdl.sri.com/projects/emerald/emerald-niss97.html.

95. Power R. (2002) *2002 CSI/FBI Computer Crime and Security Survey*, Computer Security Issues & Trends, Vol. VIII, No.1.

96. Ptacek T.H., and Newsham T.N. (1998) "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection", Technical Report, Secure Networks, January 1998, http://www.insecure.org/stf/secnet_ids/secnet_ids.html

97. Ragsdale J.D., Carver C.A. Jr., Humphries J.W., and Pooch U.W. (2001) "Adaptation Techniques for Intrusion Detection and Intrusion Response Systems", 2nd Annual IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop, West Point New York, 5-6 June 2001.

98. Richard M. (2001) "Intrusion Detection FAQ: Are there limitations of Intrusion Signatures?", SANS Institute, 5 April 2001, http://www.sans.org/resources/idfaq/limitations.php

99. Richardson R. (2003) *8th Annual CSI/FBI Computer Crime and Security Survey,* Computer Security Institute, http://www.gocsi.com/

100. Roculan J., Hittel S., Hanson D., Miller J.V., Kostanecki B., Gough J., van Velzen M., and Friedrichs O. (2003) "DeepSight Threat Management System: Threat Analysis: SQLExp SQL Server Worm Analysis", Version 2, Symantec Corp., 28 January 2003, http://securityresponse.symantec.com/avcenter/Analysis-SQLExp.pdf

101. Rogers L. (2001) "Buffer Overflows – What Are They and What Can I Do About Them?" CERT Coordination Centre, 3 December 2001, http://www.cert.org/homeusers/buffer_overflow.html

102. Russell D., and Gangemi G. T. (1991) *Computer Security Basics*, O'Reilly & Associates, Inc., Sebastopol, CA, 1991.

103. SANS Institute (2003) "The Twenty Most Critical Internet Security Vulnerabilities" Version 4.0, SANS Institute, 8 October 2003, http://www.sans.org/top20

104. Shipley G. (1999), 'Intrusion Detection, Take Two', Network Computing online journal, November 15 1999, http://www.networkcomputing.com/1023/1023f1.html?ls=NCJS_1023bt

105. Shipley G. (2003) "Security Watch: Don't Get Bitten by NIPS Hype", Network Computing, 13 June 2003, http://www.networkcomputing.com/1411/1411colshipley.html

106. Singh H. (2004) *A Correlation Framework for Continuous User Authentication Using Data Mining,* PhD Thesis, University of Plymouth, United Kingdom.

107. Snyder J. (2003) "Review: False positives remain a major problem", Network World Fusion Online Magazine, 13 October 2003, http://www.nwfusion.com/reviews/2003/1013idsalert.html

108. Stoll C. (1991) *The Cuckoo's Egg,* Pan Books, London UK.

109. Symantec (2001) *NetProwler: Dynamic Intrusion Detection for Enterprise Networks,* http://enterprisesecurity.symantec.com/content/promotions.cfm?PDFID=29&PID=11748972&EID=0

110. Symantec (2003) *Symantec Internet Security Threat Report*, Volume III, February 2003.

111. Symantec (2003b) "Symantec™ ManHunt 3.0.1: Administration Guide", Symantec Corp., http://www.symantec.com/.

112. Symantec (2004) "Symantec Security Response", http://securityresponse.symantec.com/

113. Talisker (2001) "Intrusion Detection Systems: Re: HIDS vs. NIDS market stats?" http://seclists.org/lists/ids/2001/May/0025.html

114. Talisker (2003) "Intrusion Detection Systems" http://www.networkintrusion.co.uk/ids.htm

115. Taylor S. and Wexler J. (2003) "IDS vs. IPS: Is one strategy 'better?'", Network World Fusion, 16 October 2003, http://www.nwfusion.com/newsletters/frame/2003/1013wan2.html

116. Toth T. and Kruegel C. (2002) "Evaluating the impact of automated intrusion response mechanisms", In Proceedings of the 18th Annual Computer Security Applications Conference (ACSAC), 9-13 December 2002, San Diego California, IEEE Computer Society Press, USA, http://www.infosys.tuwien.ac.at/Staff/tt/publications/Evaluating_the_Impact_of_Automated_Intrusion_Response_Mechanisms.pdf

117. UC Davis (2000) "UC Davis Response and Detection Project Overview", December 2000, http://seclab.cs.ucdavis.edu/response/

118. UK Audit Commission (2001) *Your business@risk - An update on IT Abuse 2001*, Audit Commission Publications, September 2001.

119. Wang X., Reeves D.S. and Wu S.F. (2001) "Tracing Based Active Intrusion Response", Journal of Information Warfare, Vol.1, Issue 1, September 2001: pp 50-61.

120. Wang X., Reeves D.S., Wu S.F. and Yuill J. (2001a) "Sleepy Watermark Tracing: An Active Network-Based Intrusion Response Framework", in Proceedings of the IFIP TC11 16th Annual Working Conference on Information Security: Trusted Information: The New Decade Challenge, 11-13 June, Paris, France: pp 369 – 384.

121. WWWD (2002) "Statistics for WWWD2", WorldWide WarDrive, http://www.worldwidewardrive.org/wwwd2/wwwd2stats.html

122. Zone-h (2004) "Digital Attacks Archive", 23 March 2004, http://www.zone-h.org/en/defacements

123. 3i (2002) *E-security - 2002 and beyond,* 3i Group plc, 23 January 2002, http://www.3i.com/pdfdir/3i_esecurity_2002.pdf.

LIST OF PUBLICATIONS

LIST OF PUBLICATIONS

1. Furnell S.M., Magklaras G.B., Papadaki M., and Dowland P.S. (2001) "A Generic Taxonomy for Intrusion Specification and Response", in Proceedings of Euromedia 2001, Valencia, Spain, 18-20 April 2001: pp 125-131.

2. Papadaki M., Magklaras G.B., Furnell S.M., and Alayed A. (2001) "Security Vulnerabilities and System Intrusions – The need for Automatic Response Frameworks", in Proceedings of IFIP 8th Annual Working Conference on Information Security Management & Small Systems Security, Las Vegas, 27-28 September 2001: pp 87-97.

3. Papadaki M., Furnell S.M., Lines B.M., and Reynolds P.L. (2002) "A Response-Oriented Taxonomy of IT System Intrusions", in Proceedings of Euromedia 2002, M.Roccetti (ed.), Modena, Italy, 15-17 April 2002: pp 87-95.

4. Dowland P.S., Furnell S.M., and Papadaki M. (2002) "Keystroke Analysis as a Method of Advanced User Authentication and Response", in Security in the Information Society: Visions and Perspectives, M.A.Ghonaimy et al (eds): pp 215-226.

5. Irakleous I., Furnell S.M., Dowland P.S., and Papadaki M. (2002) "An experimental comparison of secret-based user authentication technologies", Information Management & Computer Security, vol. 10, no. 3: pp 100-108.

6. Papadaki M., Furnell S.M., Lee S.J., Lines B.M., and Reynolds P.L. (2002) "Enhancing response in Intrusion Detection Systems", Journal of Information Warfare, vol. 2, no. 1, 2002: pp90-102.

7. Papadaki M., Furnell S.M., Lines B.M., and Reynolds P.L. (2003) "Operational Characteristics of an Automated Intrusion Response System", in Communications and Multimedia Security: Advanced Techniques for Network and Data Protection Lioy A. and Mazzochi D. (eds), Springer Verlang, October 2003: pp 65-75.

8. Papadaki M. and Furnell S.M. (2004) "IDS or IPS: what is best?", Journal of Network Security, July 2004:pp15-19.

9. Papadaki M., and Furnell S.M. (2004) "Automating the process of intrusion response", to appear in Proceedings of 5th Australian Information Warfare and Security Conference, Perth, Western Australia, 25-26 November 2004.

Poster presentations

1. Dowland P.S., Furnell S.M., Magklaras G.B., Papadaki M., Reynolds P.L., Rodwell P., Singh H. (2000) "Advanced Authentication and Intrusion Detection Technologies", Poster presentation at Britain's Younger Engineers in 2000, House of Commons, London, 4 December 2000.

2. Papadaki M., Furnell S.M., Dowland P.S., Lines B.M., and Reynolds P.L. (2002) "Enhancing Intrusion Response in Networked Systems" Poster presentation at

Britain's Younger Engineers in 2002, House of Commons, London, 9 December 2002.

3. Papadaki M. (2002) "Factors Influencing Automated Intrusion Response", Poster presentation at the 3rd International Network Conference (INC 2002), University of Plymouth, UK, 16-18 July 2002.

A GENERIC TAXONOMY FOR INTRUSION SPECIFICATION AND RESPONSE

S.M.Furnell, G.B.Magklaras, M.Papadaki and P.S.Dowland

twork Research Group, Department of Communication and Electronic Engineering, University of Plymouth, Plymouth, United Kingdom
e-mail: nrg@jack.see.plym.ac.uk URL: http://ted.see.plym.ac.uk/nrg

YWORDS

ısion Detection Systems, Intrusion taxonomy, ısion specification, Automated response.

STRACT

paper presents a preliminary description of an ısion taxonomy to aid the development of a generic ısion specification and response platform. Existing ısion taxonomies are assessed in order to derive a ıble classification of incidents that would be both ctable and addressable by an automated intrusion ction system. The issue of automated responses to ısions is considered, along with the factors that would ıence the level of response selected. This work esents a contribution to ongoing research in relation to Intrusion Monitoring System, a conceptual iitecture for Intrusion Detection.

RODUCTION

the last twenty years, the computer security world has ıessed the growth and continuous development of usion Detection Systems (IDS). These tools monitor events occurring in a computer system or network and ch for indications of security-related problems. There many challenges in the development process of these ems and, to date, the majority of research has centred ınd the issue of how an intrusion may be detected ıkherjee et al. 1994). One issue that has not been clusively addressed is the classification of different usions into a consistent framework that can be used as ısis for further work. With an appropriate taxonomy he core, it becomes possible to pursue related work in ıtion to both the specification of, and response to, usions.

; considered that a suitable specification of an intrusion terms of the detectable indicators) may be used as ut to an IDS to enable the identification of the ociated attack. At present, there is only one widely ognized theoretical study of intrusion specification, cribed by Feirtag et al (2000). However, the derived ımmon Intrusion Specification Language' has a number disadvantages that might limit its application to large commercial systems. It is outside the scope of this paper to systematically discuss these disadvantages but the reader can find additional reference in (Doyle, 1999). The existence of these limitations indicates strongly the need for a more systematic examination of the foundations of an Intrusion Specification Language. It is also important for recognised intrusions to be linked to appropriate responses.

The issue of *automated* response is important for the following reasons:

- there is an increasing need to ease the load on system administrators/security architects as corporate IT infrastructures become larger and more complicated.
- many intrusion incidents are generated by automated scripts. As a result, the speed with which a response should be initiated is great. Moreover, the increase in network bandwidth coupled with the distributed nature of many attacks and the exponential growth in CPU power, narrows the margins left for a non-automated system response.

Despite this, the issue of automated response has been widely neglected in the process of developing research prototypes and commercial IDS products, the focus having been given to detecting the intrusions themselves.

This paper aims to establish the foundations for developing a generic Intrusion Specification Language and response platform at a preliminary level. The discussion begins with an outline of the Intrusion Monitoring System (IMS), a conceptual architecture that represents the focus of the research to be presented. This is followed by a brief review of existing intrusion taxonomies, leading into an overview description of a derived approach, which is considered to represent a suitable basis for considering the issues of intrusion specification and response. The issue of automated response is then considered, presenting the top-level considerations for an intrusion response framework and an example of how this could be applied in practice. The paper concludes with a look ahead to intended further research in this area.

THE INTRUSION MONITORING SYSTEM

IMS is a conceptual architecture for intrusion monitoring and activity supervision, based around the concept of a centralised host handling the monitoring of a number of networked client systems. Intrusion detection in the system is based upon the comparison of current user activity against both historical profiles of 'normal' behaviour for legitimate users and intrusion specifications of recognised attack patterns. The architecture is comprised of a number of functional modules, addressing data collection and response on the client side and data analysis and recording at the host. The roles of these modules are summarised below.

The **Anomaly Detector** analyses the data gathered by the IMS clients for signs of suspected intrusion. This data can be compared against both the user's behaviour profile and the generic intrusion specifications (i.e. attack signatures).

The **Profile Refiner** allows the automatic modification of a user's profile in response to a valid session profile. This recognises the fact that a user's behaviour pattern may change over time.

The **Recorder** stores a temporary record of system and user activity during a session (session profile) which can be used by the Profile Refiner to update the user profile, providing the session was not considered anomalous.

The **Archiver** provides an audit log, storing all security relevant events.

The **Collector** provides an interface between the IMS client and the applications running on the client computer. The collector is responsible for gathering information relevant to the user and system activities.

The **Responder** provides the interface between the IMS software suite and the end-user. Its main task is that of monitoring the signals sent from the server to the client and taking appropriate action where necessary. This will be considered further in the sections that follow.

The **Communicator** provides the interface between the client and server IMS software. The communicator is responsible for ensuring a consistent, reliable and secure exchange of data between the client and server.

The **Controller** provides a management interface, allowing an administrator to configure the IMS system-operating parameters.

The architecture is described in more detail by Furnell and Dowland (2000). For the purposes of the discussion in this paper, the key elements are the anomaly detector (which would make use of appropriate intrusion specifications derived from the taxonomy) and the responder (which deals with suspected problems).

EXISTING INTRUSION TAXONOMIES

The first step towards establishing an Intrusion Specification Language (ISL) is to derive a taxonomy of intrusive activities. A number of intrusion taxonomies have been devised to date. However, before these are considered, it is useful to define the terms 'intrusion' and 'intrusion taxonomy'. Appropriate definitions are provided by Amoroso (1999), who defines the term intrusion in an IT context as *"a sequence of related actions by a malicious adversary that results in the occurrence of unauthorized security threats to a target computing or networking domain"*. The reader will notice that this definition emphasises the existence of a set of resources, dividing them into computers and networking (telecommunication equipment that interconnects the discrete computing units). The author proceeds further and defines the term intrusion taxonomy to be a *'structured representation of intrusion types that provides insight into their perspective relationships and differences'*. In this case, the author denotes the process of spotting common or major differences between intrusions as a measure to ease the automation of a response.

There are currently, there are three widely accepted intrusion taxonomies. A brief overview of these is given below.

- **SRI Neumann-Parker Taxonomy** (Neumann and Parker, 1989): An intrusion taxonomy based on a large number of incidents reported to the Internet risks forum. The taxonomy classifies intrusions into nine categories, according to key elements that might indicate a particular type of incident. Table 1 below summarises the overall scheme.

Table 1: SRI Neumann-Parker (NP) Taxonomy

NP 1 EXTERNAL MISUSE	Non-technical, physically separate intrusions
NP 2 HARDWARE MISUSE	Passive or active hardware security problems
NP 3 MASQUERADING	Spoofs and identity changes
NP 4 SUBSEQUENT MISUSE	Setting up intrusion via plants, bugs
NP 5 CONTROL BYPASS	Going around authorised protections/controls
NP 6 ACTIVE RESOURCE MISUSE	Unauthorised changing of resources
NP 7 PASSIVE RESOURCE MISUSE	Unauthorised reading of resources
NP 8 MISUSE VIA INACTION	Neglect or failure to protect a resource
NP 9 INDIRECT AID	Planning tools for misuse

Lindqvist and Jonssen's intrusion taxonomy (Lindqvist and Jonsson, 1997): This effort could be considered as an extension of the SRI Neumann-Parker taxonomy. It further refines security incidents into intrusions, attacks and breaches. It examines these issues from a system-owner point of view, based on a number of laboratory experiments. The results of these experiments indicated a need for further subdivision of the Neumann-Parker classes 5, 6 and 7, as shown in table 2 below. Their work provides further insight into the process of spotting aspects of system elements that might indicate an intrusion.

Table 2: Lindqvist and Jonssen Extension of the SRI NP Taxonomy

Extended NP5 CONTROL BYPASS	Password attacks, spoofing privileged programs, utilising weak authentication
Extended NP6 ACTIVE RESOURCE MISUSE	Exploitation of write permissions, resource exhaustion
Extended NP7 PASSIVE RESOURCE MISUSE	Manual browsing, automated browsing

- **John Howard's security incident analysis** (Howard, 1995): This is focused on the process of attack, rather than classification categories. It establishes a link through the operational sequence of *tools*, *access*, and *results* that connects the attackers to their objectives. Although Howard's work cannot be considered as a pure taxonomy, the wealth of statistical analyses and the various cases mentioned provides some of the most well-written and useful material for considering/revising new taxonomies.

A PROPOSED TAXONOMY FOR INTRUSION SPECIFICATION AND RESPONSE

Although the previously mentioned taxonomies are indeed useful for the systematic study of intrusions, none of them is tailored for the purposes of producing the structure of an Intrusion Specification Language. The classification criteria employed by these taxonomies cannot be qualified or quantified very easily by an Intrusion Detection System. The best way to overcome this problem is to devise an intrusion taxonomy scheme that is based on elements of the IT infrastructure that are being targeted. The idea is that it is easier to detect which particular element is affected by an intrusive action, rather than trying to sense the origin, entity or the motives for initialising an attack. This information is also considered sufficient to determine the main options for response. As a consequence, the following target-based intrusion classification schema has been devised, based on things that could be directly detected by an Intrusion Detection System (IDS). The level of IT component granularity increases towards the bottom layers of the suggested hierarchy, all the way down to individual self-contained components. This level of granularity is necessary for devising a comprehensive Intrusion Specification Language set. However, the language itself will not be defined in this paper and, as such, the discussion will consider only the top three layers of the suggested taxonomy.

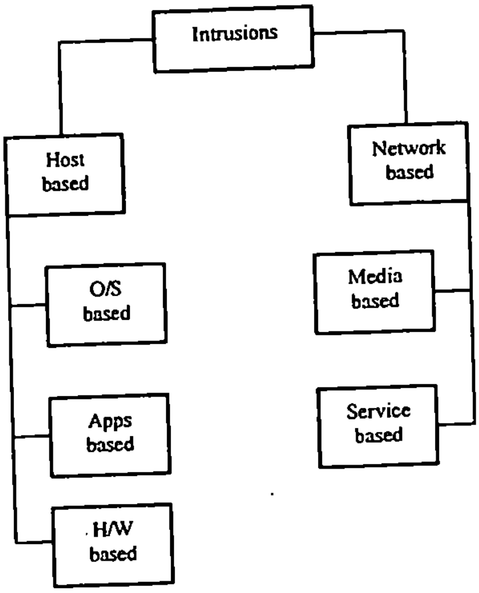


Figure 1: Levels 1 and 2 of the Taxonomy

Figure 1 indicates that, at the top level, intrusions can be sub-divided into host and network based categories. This is because certain attacks focus upon computing systems (servers, desktop workstations, thin/embedded clients), whilst there are others that target the equally important elements that interconnect them.

The host-related intrusions are divided into three major sub-categories. The operating system (O/S) based category includes all intrusive activities that aim to compromise functions such as memory management, I/O activity and file storage operations (see Figure 2). A typical example of a host-related attack could be a buffer overflow attack.

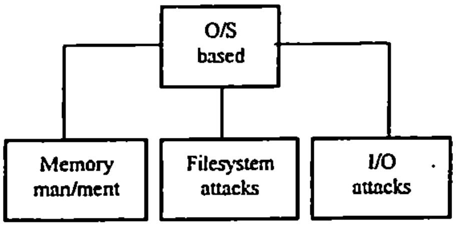


Figure 2: Operating System Intrusions

The application-based intrusion category concerns all intrusions that may affect the operation of a particular software package that is using the various operating system services, as described in Figure 2. However, this category refers specifically to files that are maintained by the application itself, rather than generic system or user data files. These files often carry a particular extension and could be manipulated in various ways in order to halt or affect the operation of the application in specific ways. For example, if a configuration file of the application is changed, then it is possible to make the application disclose confidential information. If an application log (data) file is manipulated, then valuable data might be lost or stolen (Figure 3). Although there is a substantial overlap between application and operating system intrusions, the two should not be confused. For instance, if a non-legitimate user modifies an application file, then the problem is really related to the failure of the Operating System to authenticate the file manipulation. However, if this action is initiated by a legitimate user, then the application itself should contain additional functionality to detect and contain the resulting effects and the incident should belong to the application-based category of our taxonomy.

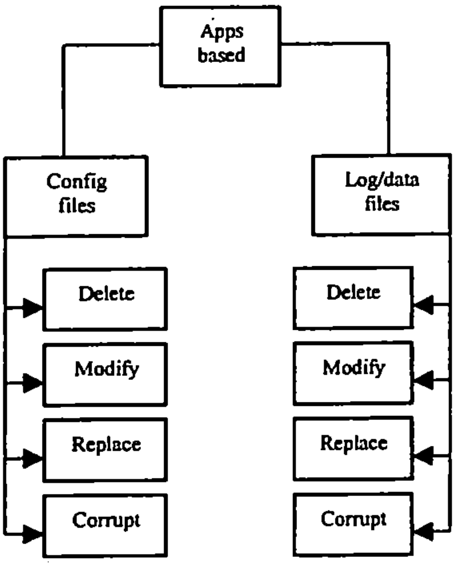


Figure 3: Application-based Intrusions

Finally, intrusive activities may concern the hardware components of a host. For instance, the non-authorised addition of a modem on a secure server may or may not provide a security threat because it opens the door to a non-secure environment such as the Public Switched Telephone Network (PSTN). Theft, vandalism and changes in the configuration of hardware components, in order to disable security features are also common scenarios, illustrated in Figure 4.

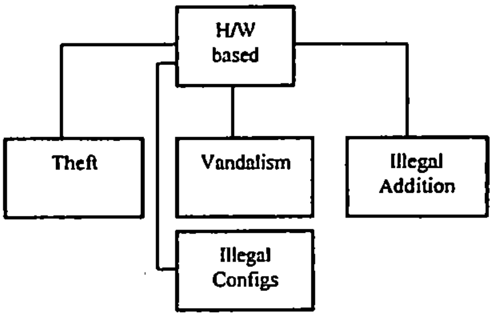


Figure 4: Hardware-based Intrusive Activities

Network-related intrusions could be further subdivided into media and serviced-based intrusions. The word 'media' encompasses all the hardware components that are responsible for the physical transfer of the network packets, whereas 'services' are discrete functions performed by specific telecommunication elements such as routers, gateways, firewalls and other devices.

In line with what can happen with host related hardware, media can be stolen, vandalised or configured in a non-authorised way. In addition, many intrusive activities tend to target the physical signaling of the medium itself, something that is not common in host-related hardware. The detection of these disruptions is still a fruitful area of research.

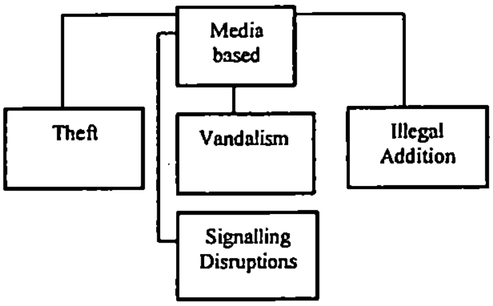


Figure 5: Network Media-based Intrusions

Finally, service-based attacks might target the smooth operation of routing and management services. The former concerns the vital operation of network equipment: without routing no network can function. The latter is also important for the smooth operation of large corporate data networks and concern tools that configure, troubleshoot and provide redundancy services (network address translation, load balancing).

As previously indicated, this classification provides a fairly high level view, but it is sufficient to begin classifying practical incidents and determine appropriate responses. For the detection of a particular intrusion, a more precise specification is necessary, requiring further levels of decomposition within the taxonomy.

AUTOMATED RESPONSES IN INTRUSION DETECTION SYSTEMS

Intrusion response can be defined as the process of counteracting the effects of an intrusion. It includes the series of actions taken by an Intrusion Detection System, which follow the detection of a security-related event. It is important to note that consideration is not only given to taking action after an intrusion has been detected, but also when events of interest take place and raise the alert level of the system. That is the early stages of an attack, when the system is suspecting the occurrence of an intrusion, but is not yet confident enough to raise an alarm.

The aims of response actions can be summarised into the following:

1. Protect system resources
   - in the short term, this will include mechanisms to contain the intrusion, as well as to recover and restore the system to a well known state
   - in the longer term, learn from the intrusion and use this knowledge to remove identified vulnerabilities of the system, and to enhance the detection and response capability. The underlying idea is to make sure that the intrusion cannot be repeated.

2. Identify the perpetrator of the intrusion.

The contribution of automated response can be mostly focused on the protection of system resources. Further investigation of the intrusion to identify the perpetrator is thought to require co-operation with other parties, like Incident Response Teams, and mostly falls under the operational aspect of response.

Issues in automated response

One of the issues we need to consider for response to intrusions is the confidence level of the system, in order to avoid false alarms. In the case of a false positive, we may find automated response itself to become a denial of service issue, by affecting the access level of legitimate users. Recommended actions to increase the certainty level are based on a combination of detection and reaction in order to collect additional information about the attack. According to the level of confidence and the seriousness of the potential intrusion, those actions could be:

- further investigate details of the intrusion in audit log files;
- record details in an intrusion log for further inspection / investigation;
- alert the system administrator and increase the intrusion alarm;
- increase the monitoring level;
- issue a challenge for further authentication;
- limit permitted user behaviour;
- delay (or lower priority of) intruder's session / process;
- terminate (or suspend) the anomalous session / process.

The severity, as well as the discrete characteristics of an intrusion, are also issues that need to be matched to the confidence level, to determine and prioritise actions of response. It is important to recognise and identify the threats posed to the system so that appropriate actions can be taken in time, to prevent the system from reaching a compromised state.

Furthermore the impact of response actions upon users and the system is another consideration that should also be taken into account. It is desirable to preserve the transparency of system response as much as possible, so that no disturbance to legitimate users will be added and no alert to attackers will be given to make them aware of the fact that they are being monitored. The latter might give attackers the opportunity to cover the traces of their activities, and possibly cause further damage to the system. Alternatively, the sooner actions are taken, the safer it is for the system to preserve its state and minimise the damage from the attack.

The overall process is illustrated in Figure 6, which indicates the inputs to an entity such as the IMS Responder and shows the possible responses that may be taken at different impact levels.

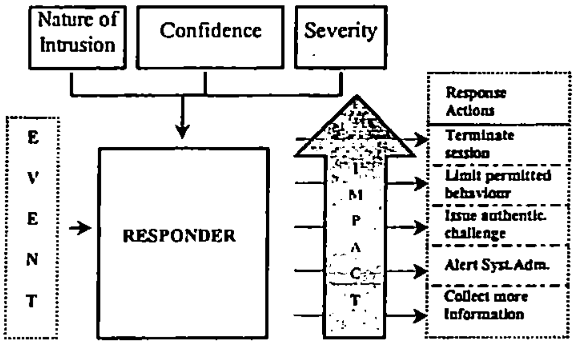


Figure 6 : Issues in Response

Example - Counteracting DoS attacks

As an example of potential response levels, this section considers the issue of Denial of Service (DoS) attacks – which would be classified as network/service-based intrusions under the earlier taxonomy. DoS attacks are an increasing threat to Internet systems, as illustrated by the fact that they account for 60% of reported incidents affecting WWW sites (Power, 2000).

Speed of detection and response is a major requirement for this class of attack. They are difficult to guard against - mainly due to the fact that they are identifiable from their results (i.e. when it is already too late to prevent them). The issue of how to respond to DoS attacks is an area of ongoing work in the research community. The most dominant approach is *resource management*, which is based on monitoring the resource requirements of computational tasks, adjusting their priorities to make sure that the capacity of the resource is not overloaded. It may include resource management for both the host and network domain, defining intra-host parameters (scheduling, storage management) and inter-host channels of allocation (task migration, network flow control) (Tung, 1997).

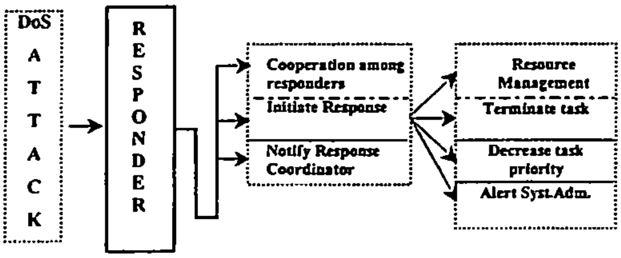


Figure 7: Response Actions for DoS Attacks

However, resource management may not be the only, or most desirable, response in any given situation. Examples of different levels of response that may be taken against a DoS attack are illustrated in Figure 7, which also indicates the stages that a Responder agent may take in a networked monitoring environment in order to mount a co-ordinated response.

CONCLUSIONS AND FUTURE WORK

The taxonomy presented in this paper provides the foundation for ongoing work in relation to the issues of intrusion specification and response.

A generic Intrusion Specification Language will be based around a full version of the taxonomy presented in this paper and will enable the description of events in a manner that is independent of particular system / application configurations. It is intended that the language will facilitate the description of both an attack and the consequent response that should be applied.

The response framework is also the focus of ongoing research. The main tasks will involve classifying the range of responses appropriate to the different categories of intrusion from the taxonomy, and then measuring the effectiveness of the different actions (considering their impact to the system/legitimate users and strength against attackers).

It is considered that cooperation between Responders residing in different networks would be a desirable feature. Coordination of those elements will then be needed and the evaluation of possible response strategies will be examined. A possible disadvantage of this approach would be the utilisation of this feature to deceive responders and utilise them either as information sources or agents to launch attacks. Thus careful consideration should be given for the secure communication between those elements.

REFERENCES

Amoroso E. 1999. 'Intrusion Detection: An Introduction to Internet Surveillance, Correlation, Traps, Trace Back, and Response' , Second Printing, Intrusion.Net Books, New Jersey, June 1999, Chapter 4, pp100-105.

Doyle, J. 1999. "Some representational limitations of the Common Intrusion Specification Language (CISL)", http://www.medg.lcs.mit.edu/projects/maita/documents/cc2/cisl/

Feirtag R., Kahn C., Porras P., Schnackenberg D., Staniford-Chen S., Tung B. 2000. 'A Common Intrusion Specification Language'. http://www.gidos.org/

Furnell, S.M. and Dowland, P.S. 2000. "A conceptual architecture for real-time intrusion monitoring", *Information Management & Computer Security*, vol. 8, no. 2: pp65-74.

ward, J. 1997. *An Analysis of Security Incidents on the Internet 1989 – 1995*. Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, April 1997. http://www.cert.org/research/JHThesis

ndqvist U. and Jonsson E. 1997. "How to Systematically Classify Computer Security Intrusions", *Proceedings of the 1997 IEEE Symposium on Security and Privacy*, May 4-7, 1997, IEEE Computer Society Press.

ukherjee, B.; Heberlein, L.T.; Levitt, K.N. 1994. "Network Intrusion Detection", *IEEE Networks 8*, no.3: 26-41.

eumann, P.G. and Parker, D.B. 1989. "A summary of computer misuse techniques". In *Proceedings of the 12th National Computer Secuirty Conference* (Balitimore, USA, 10-13 Oct): pp396-407.

wer, R. 2000. "2000 CSI/FBI Computer Crime and Security Survey", *Computer Security Issues and Trends*, Vol. V1, No. 1. pp1-15.

ng, B. 1997. "CRISIS: Critical Resource Allocation and Intrusion Response for Survivable Information Systems", Presentation held at *Intrusion Detection Workshop* (Savannah GA, February 1997). See http://www.isi.edu/~brian/crisis/inprint/savannah.ps

OGRAPHIES

EORGE MAGKLARAS has gained a first class onors degree in Computer Systems & Networks from e School of Computing, University of Plymouth. He has worked as a network and software development specialist at the European Headquarters of the IBM NUMA-Q team in England, where he specialised on operating system and networking support for Symmetric Multi-Processing (SMP) servers. He has also consulted on a number of UNIX/LINUX IT projects in England, Greece and the Netherlands. He has won a three-year EPSRC studentship to pursue a PhD on a 'Generic architecture for Intrusion Specification and Misuse Detection in IT systems'. His work is also supported by the Metropolitan Police Computer Crime Unit and Orange Personal Communications. He is currently working as a researcher and part time lecturer with the Network Research Group at the University of Plymouth.

MARIA PAPADAKI was born in Iraklio of Crete, Greece and studied Informatics in the Technological Educational Institute (T.E.I.) of Athens. After her graduation in November 1997, she worked for two years for the Library and the Network Operating Centre of the Athens School of Fine Arts. Funded by the State Scholarships Foundation (SSF) of Greece, she attended the MSc course *Integrated Services and Intelligent Networks Engineering* at University of Plymouth, UK (1999-2000) and is currently a research student within the Network Research Group of the University. Current interests include intrusion detection and methods of automated system response.

SECURITY VULNERABILITIES AND SYSTEM INTRUSIONS

The need for Automatic Response Frameworks

S.M.FURNELL, M.PAPADAKI, G.MAGKLARAS and A.ALAYED
Network Research Group,
Department of Communication and Electronic Engineering,
University of Plymouth,
Plymouth,
United Kingdom,
Tel: +44 1752 233521,
Fax: +44 1752 233520,
Email : nrg@jack.see.plym.ac.uk

Key words: Vulnerability analysis, intrusion detection, intrusion response

Abstract: Addressing security vulnerabilities and system intrusions can represent a significant administrative overhead in current computer systems. Although technologies exist for both vulnerability scanning and for intrusion detection, the problems typically require some form of human intervention before they can be rectified. Evidence suggests that, in many cases, this can lead to omissions or oversights in terms of protection, as administrators are forced to prioritise their attention to security amongst various other tasks (particularly within smaller organisations, where a dedicated security administration function is unlikely to be found). As a result, mechanisms for automated response to the issues are considered to be advantageous. The paper describes the problems associated with vulnerability analysis and intrusion response, and then proceeds to consider how, at a conceptual level, the issues could be addressed within the framework of a wider architecture for intrusion monitoring.

1. INTRODUCTION

The widespread use of Internet systems by organisations of all types means that the problem of IT security has never been more prominent. It would be no exaggeration to say that many organisations and individuals are reliant upon these systems, their correct operation and the data they contain. Despite their critical role, however, evidence has shown that systems are often vulnerable to various forms of abuse – breaching their security and resulting in intrusions. The problem of security breaches has substantially increased in recent years. In the CSI/FBI 2000 Computer Crime and Security Survey, financial losses due to computer security breaches mounted to $377,828,700, while the average annual total over the three years prior to 2000 was $120,240,180 [1].

An *intrusion* is the series of actions taken by an attacker against a target to achieve an unauthorised result. In order to fulfill this objective, the attacker must exploit a computer or network *vulnerability*, which represents the weakness of the system that allows unauthorised action to be taken [2]. For example, a well-known system vulnerability is the use of weak, default or even blank passwords [3]. These offer the opportunity for effortless access by attackers, who will routinely attempt to gain access to systems by trying default passwords, and then easily guessable ones. Only if these are unsuccessful will they need to resort to more sophisticated methods. Once inside, attackers can exploit other widely known vulnerabilities to increase their access (e.g. to attain root / administrator privileges).

This paper considers the dual problems of addressing security vulnerabilities and responding to intrusions that may result from their exploitation. In current systems, both elements can be seen to represent an administrative burden, with responsibility falling to system administration staff. In many cases, this may lead to omissions and prioritisation problems, as the same staff will often have numerous other responsibilities. It is considered that this issue is likely to be particularly acute within smaller organisations, due to the typical lack of dedicated IT security management staff. The discussion begins with an examination of the administrative problems posed by security vulnerabilities, in terms of the efforts required to identify and resolve an ever-increasing range of known problems. It then proceeds to consider the further considerations involved if it becomes necessary to respond to a suspected intrusion incident – which will often result from the exploitation of a vulnerability. The desirability of automated responses is recognised in both cases, leading to consideration of how an automated framework could be used to reduce the burden upon system administrators.

2. THE ADMINISTRATIVE PROBLEM OF SECURITY VULNERABILITIES

It is recognised that responding to both security vulnerabilities and detected intrusions can represent a significant administrative overhead. In the case of vulnerabilities, for example, there are associated overheads at two levels:

a) ensuring awareness of vulnerability existence;
b) being able to take appropriate corrective action to resolve them (e.g. installing software upgrades and patches).

Even though many exploits are based upon vulnerabilities that have been known for some time, the problem is a difficult one to keep on top of. Many software developers routinely release patches that enable known bugs and vulnerabilities in their products to be rectified – in some cases this happens before particular weaknesses have become publicly known, whilst in others it is in response to a problem being reported. As a result, the situation in many cases is that simple maintenance activity by system administrators is all that would be required to plug the holes. However, despite this, the problems clearly remain. The SANS Institute has identified several reasons why this may be the case [4]:

- 1.2 million new computers are added to the Internet every month;
- there is a lack of security experts to address the problems;
- the number of vulnerabilities continues to grow and there is no priority list for dealing with them.

From the system administrator's perspective, the main requirement is to ensure that the system remains operational and available – this is what the users expect and complaints will quickly occur if this is not the case. So, unless installing a patch is explicitly required to ensure that this is the case, then the task is likely to be given a lower priority.

Looking at the number of warnings that are issued, it is easy to see how administrators might downgrade the importance of responding to them immediately. This can be illustrated by considering the security bulletins issued by Microsoft Corporation in relation to its product range. When vulnerabilities are identified in Microsoft products, the company works to develop a solution and then issues an advisory bulletin when a software patch or upgrade is available for download. The graph in *Figure 1* summarises the number of security bulletins issued per month, between January 1999 and September 2000 (statistics obtained from http://www.microsoft.com/technet/security/current.asp).

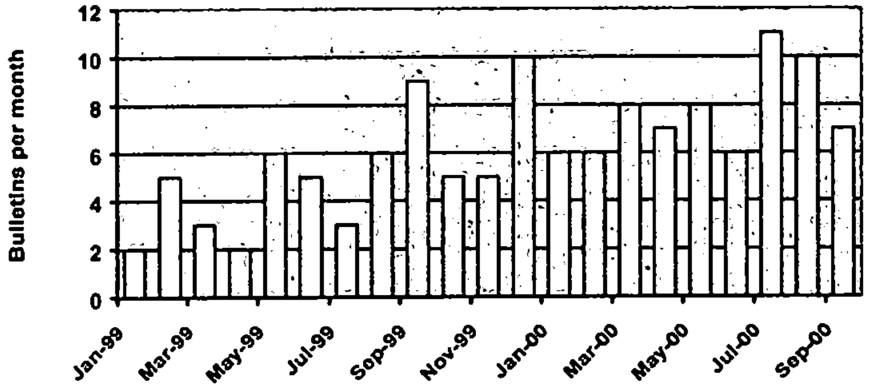


Figure 1. Microsoft Security Bulletins (January 1999 to September 2000)

It can be seen from the graph that the number of security bulletins issued ranges from two per month up to eleven per month (the average was 6 per month over the 21 month period). This might not be so bad if the associated patch was being installed on just a single system, but in some cases an organisation's IT and network configuration may dictate that the administrator must go around and update a number of individual systems in turn (which could obviously become quite time consuming). In some cases, the number of systems may run into the thousands, whereas the administration team may number less than ten. Relating this to the number of patches released per month, this could lead to each administrator having to patch about 20 machines per day (assuming the average of 6 patches per month and that all systems required them). It should also be remembered that these bulletins are only those related to Microsoft products. Where an organisation's IT set up is based upon a heterogeneous, multi-vendor configuration, security advisories from other sources would also have to be taken into consideration.

So, in view of all this, it can be appreciated that administrators might start out with good intentions, responding to each advisory as it arrives. However, this could quickly become burdensome and so the decision may be taken to batch them up and respond to them on a less frequent basis. Whilst this makes good administrative sense, it is less sensible from a security perspective. Once an advisory has been issued, the information about the associated vulnerability is available to anyone – and any hackers who were not aware of it before will certainly have access to it from then on. As such, any systems in which the weakness has not been addressed are exposed to a greater level of risk than before the advisory was made.

So what is the effect of not installing the available fixes? According to Attrition.org, 99% of the 5,823 web site defacements that occurred during 2000 were as a result of failure to patch known vulnerabilities for which the fixes were already available [5].

3. INTRUSION RESPONSE

If a vulnerability is successfully exploited, a system intrusion is likely to result – which will require some form of consequent response. From this perspective, the issues of vulnerability analysis and intrusion response are related areas, separated only by the occurrence of an incident.

Intrusion response can be specified as the process of counteracting the effects of an intrusion. It includes the series of actions taken by an Intrusion Detection System, which follow the detection of a security-related event. It is important to note that consideration is not only given to taking action after an intrusion has been detected, but also when events of interest take place and raise the alert level of the system. That is the early stages of an attack, when the system is suspecting the occurrence of an intrusion, but is not yet confident enough.

It is possible to distinguish two main approaches to intrusion response, namely human/organisational approaches and technical methods. The former are those that involve human processes and organisational structures, and may include actions such as reporting an incident to the police or invoking disciplinary procedures (e.g. in cases where internal personnel are responsible). By contrast, technical responses involve the use of functional techniques and software-based methods. These technical actions can themselves be further sub-classified, into either passive or active forms of response [6]:

- **Passive responses**: aim to notify other parties (administrators - users) about the occurrence of an incident, relying on them to take further actions about it. Alarms, notifications and SNMP Traps are the most common passive responses. Passive actions are the most common response options in commercial IDS systems.
- **Active responses**: are the actions taken by a process or system to encounter the incident that has occurred. Those actions might include collecting more information about the incident, limiting permitted user behaviour, or blocking IP traffic through firewalls and routers.

Within these categories there are myriad individual response actions that could be pursued and some decision making ability is required when a suspected incident presents itself. However, although the type of incident will suggest a range of possible responses, the classification of incident alone does not provide enough information to determine which one(s) are actually appropriate. The *specific* response(s) to initiate will depend upon a number of factors, which collectively identify the context in which the incident has occurred. This idea is illustrated in *Figure 2*.

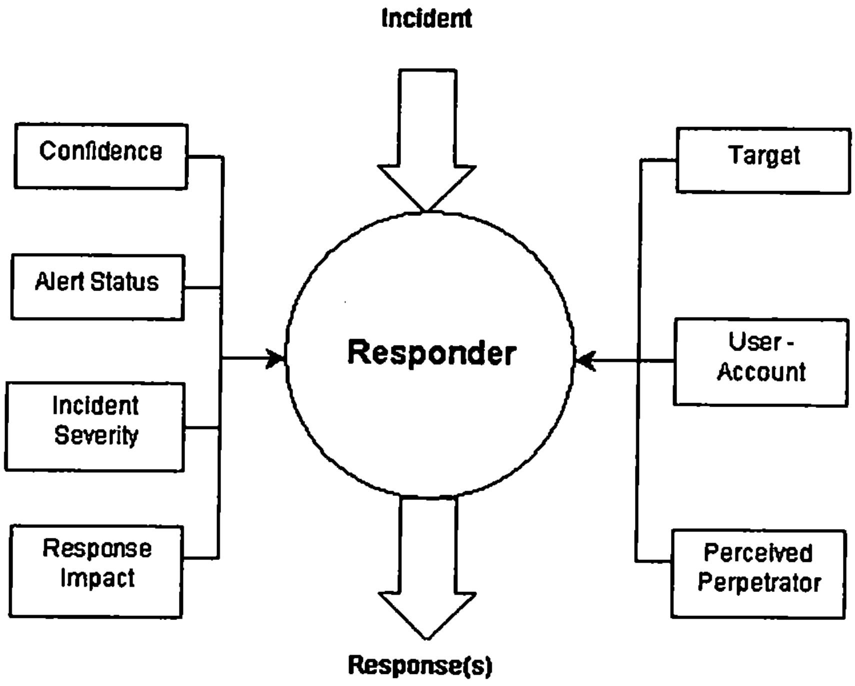


Figure 2. Factors influencing intrusion response

As the diagram shows, the *incident* is the trigger for the response and still represents the principal influence over what should be done. However, the other influencing factors that also need to be considered are as follows:

- **Confidence:** how many monitored characteristics within the system are suggestive of an intrusion having occurred?
- **Alert status:** what is the current status of the monitoring system, both on the suspect account / process and in the system overall?
- **Incident severity:** what impact has the incident already had upon the confidentiality, integrity or availability of the system and its data? How strong a response is required at this stage?

- **Response Impact:** what would be the impact of initiating a particular form of response? How would it affect a legitimate user if the suspected intrusion was, in fact a false alarm? Would there be any adverse impacts upon other system users if a particular response action were taken?
- **Target:** what system, resource or data appears to be the focus of the attack. What assets are at risk if the incident continues or is able to be repeated?
- **User account:** if the attack is being conducted through the suspected compromise of a user account, what privileges are associated with that account?
- **Perceived perpetrator:** does the evidence collected suggest that the perpetrator is an external party or an insider?

At the heart of *Figure 2* was an entity referred to as the *responder*. This is the element that will assess the various factors in order to select and invoke the required response(s). Although a great deal of work has been done in the area of automated intrusion detection, current systems are able to do very little in terms of automated response when they suspect a problem. So, in current systems, the responder role is likely to be taken by a system administrator. However, there are practical limits to the effectiveness of this approach. Firstly, the administration of increasingly large and complicated IT infrastructures becomes correspondingly more cumbersome. Secondly, the widespread use of automated scripts to generate attacks of a distributed nature [7] can render the speed of traditional response methods inadequate. As with vulnerability analysis and resolution, therefore, the administrative burden may again mean that the handling of intrusion response becomes sidelined - although, of course, there may be more incentive to respond to an intrusion because it represents a vulnerability that has already been exploited.

4. AUTOMATED RESPONSE FRAMEWORKS

In order to assist in resolving the problem of administrative overhead, some form of automated response framework is desirable. For vulnerabilities, it can be observed that there are already numerous tools available to assist in the task of scanning systems to identify potential holes. However, this only goes part of the way to addressing the problem. It relieves the administrators of having to have the detailed knowledge of system security necessary to identify weaknesses, but it still requires their attention to both run an analysis and take consequent corrective actions.

Although some scanning software includes functionality for fixing problems identified, the current approaches are limited - minor system configuration weaknesses can be rectified, but many vulnerabilities require more substantial action than this. Given that vulnerabilities and intrusions are linked issues, it makes sense for vulnerability analysis and resolution to form part of an overall intrusion monitoring approach.

Figure 3 illustrates the conceptual architecture of the Intrusion Monitoring System (IMS), a research prototype that the authors are currently developing. IMS is an architecture for intrusion monitoring and activity supervision, based around the concept of a centralised host handling the monitoring of a number of networked client systems. Intrusion detection in the system is based upon the comparison of current user activity against both historical profiles of 'normal' behaviour for legitimate users and intrusion specifications of recognised attack patterns. The architecture is comprised of a number of functional modules, addressing data collection and response on the client side and data analysis and recording at the host.

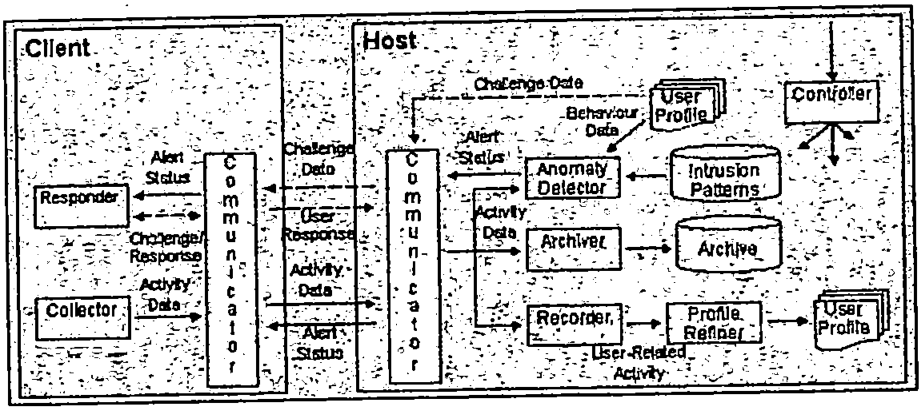


Figure 3. The Intrusion Monitoring System architecture

The full architecture is described in [8] but, from the perspective of the current discussion, the relevant modules are the *Collector*, *Anomaly Detector* and *Responder* – which can be used to perform activity monitoring (to identify intrusions) and vulnerability scanning, as well as appropriate follow-up actions in the event of problems.

The *Collector* is responsible for obtaining information from individual monitored client systems. In terms of activity monitoring, this information may relate to user data such as applications and files accessed, keystroke data (for biometric analysis) and resource usage statistics. From the perspective of vulnerability scanning, the *Collector* could also take on the role of obtaining system configuration details and the like, which would then be sent for subsequent analysis.

The *Anomaly Detector* resides on the host side and is the main recipient of the *Collector's* data. For user activity, it compares the information against historical profiles of 'normal' behaviour (e.g. frequently used applications, typing style) to identify anomalies that may indicate either an impostor or misuse by a legitimate user. In addition, generic intrusion specifications will be used to compare activities against known patterns of misuse – with a match triggering some form of alert. From a vulnerability analysis perspective, the *Anomaly Detector* will compare the collected scan data against a database of known weaknesses. In the event of problems, the *Anomaly Detector* will increase the alert status of the monitoring system and interact with the *Responder* module.

The *Responder* provides an automated facility for dealing with suspected problems. There are numerous forms of response that it would be possible to allow a system to initiate under automatic control. A small selection of ideas are listed below:

- further investigation of the incident via data collected in audit log files;
- increasing the level monitoring and/or auditing;
- issuing a challenge for further authentication;
- limiting permitted user behaviour;
- delaying (or lowering priority of) intruder's session / process;
- termination (or suspension) of the anomalous session / process.

It is the *Responder* that would be responsible for assessing and weighting the contextual factors that would determine the appropriate response option(s) for a given incident occurrence. As such, the *Responder* (like the *Anomaly Detector*) requires an element of intelligent analysis and decision-making.

In the vulnerability analysis context, the decision about what to do is potentially clear-cut, but the issue remains about when to do it. The *Responder* could conceivably take the role of coordinating and conducting updates on the affected client systems in order to resolve problems identified. A library of fixes, updates and patches would be accumulated and maintained on the host side and then issued to clients as necessary.

The description presented here proposes the solution at a conceptual level only. In practice, of course, the associated mechanisms would be far more involved and elements represented as single boxes or flows within *Figure 3* would potentially be realised as a large number of sub-processes. Some issues, such as how the system can maintain awareness of new vulnerabilities and acquire associated patches, remain unresolved and require further investigation. Other aspects, such as the anomaly detection methods and response framework, are already the focus of active research.

5. CONCLUSIONS

Automated response approaches such as those described have the potential to significantly reduce the burden on system administrators. Indeed, within the framework of an approach such as that proposed with IMS, the whole process of intrusion prevention, detection, response and resolution could be addressed.

Although the proposed approaches have the advantages identified, it is recognised that there is also a risk that any automated action taken could be incorrect. In the case of vulnerabilities, attempts to rectify security weaknesses or install software patches on the fly could adversely affect the operation of the system and/or cause incompatibility with existing elements. In the case of intrusion response, the automatic invocation of an inappropriate method could result in insufficient action being taken or, alternatively, could interrupt or deny service to a legitimate activity. As such, both are aspects that require careful configuration and their degree of permitted autonomy would strongly depend upon the nature of the system they were protecting.

The design of the automated response frameworks is the focus of ongoing research by the authors. Further details of the associated architectural approaches and implementation experiences will be reported in future publications.

6. REFERENCES

[1] CSI. 2001. "Financial losses due to Internet intrusions, trade secret theft and other cyber crimes soar", CSI Press Release, 12 March 2001. http://www.gocsi.com/prelea_000321.htm

[2] Howard, J. 1997. "An Analysis of Security Incidents on the Internet 1989 - 1995", PhD thesis. Carnegie Mellon University, April 1997. http://www.cert.org/research/JHThesis

[3] SANS Institute. 2001. "How To Eliminate The Ten Most Critical Internet Security Threats. The Experts' Consensus", Version 1.32, 18 January 2001. http://www.sans.org/topten.htm.

[4] Noack, D. 2000. "The Back Door Into Cyber-Terrorism", APBnews.com Report, 2 June 2000.

[5] CNET. 2001. “Patchwork Security - Software “fixes” routinely available but often ignored”, CNET News.com report. 24 January 2001. http://news.cnet.com/news/0-1007-201-4578373-0.html

[6] Bace, R. and Mell, P. 2001. "NIST Special Publication on Intrusion Detection Systems", National Institute of Standards and Technology (NIST), http://csrc.nist.gov/publications/drafts/idsdraft.pdf, February 12 2001.

[7] Cheung, S. and Levitt, K.N. 1997. “Protecting Routing Infrastructures from Denial of Service Using Cooperative Intrusion Detection”, Proceedings of the New Security Paradigms Workshop, Langdale,Cumbria UK, September 23 - 26, 1997, http://riss.keris.or.kr:8080/pubs/contents/proceedings/commsec/283699/

[8] Furnell, S.M. and Dowland, P.S. 2000. "A conceptual architecture for real-time intrusion monitoring", *Information Management & Computer Security*, Vol. 8, No. 2, pp65-74.

A RESPONSE-ORIENTED TAXONOMY OF IT SYSTEM INTRUSIONS

M.Papadaki[†], S.M.Furnell[†], B.M.Lines[†] and P.L.Reynolds[‡]

[†] Network Research Group, Department of Communication and Electronic Engineering, University of Plymouth, Plymouth, United Kingdom
[‡] Orange Personal Communications Services Ltd, St James Court, Great Park Road, Bradley Stoke, Bristol, United Kingdom.
e-mail: nrg@plymouth.ac.uk
Web: http://www.plymouth.ac.uk/nrg

KEYWORDS

Intrusion Detection Systems, Intrusion Taxonomy, Intrusion Response.

ABSTRACT

The ability to select and initiate appropriate response(s) is an issue that is often neglected in Intrusion Detection Systems (IDS). In order to address the problem, a means is required to consider different potential security breaches, the differing contexts in which they may occur, and the differing potential consequences. Current intrusion taxonomies have limited application in this regard, considering categories of intrusions that could not be detected by an IDS, or representing potential results in too few dimensions to enable any fine-grain selection of response options. This paper presents an overview of a new taxonomy, which is specifically targeted towards enabling the consideration of responses. A number of generic incident and target categories are identified, encompassing the most common forms of intrusion/attack and the contexts in which they may occur. An assessment of the likely results is then presented in each case, considering the security impacts, the time available to respond, and further potential attacks that may be initiated as a result. By encompassing alternative targets, and considering multi-dimensional results, the taxonomy provides a means of differentiating the incidents on the basis of the responses they require, rather than by characteristics of the attack method or their security impacts alone.

INTRODUCTION

Intrusion Detection has been an active research area within the computer security domain for more than 15 years. The challenges associated with this area have so far been largely concentrated on the process of detecting an intrusion. However, automation of the next stage after detection, the response to an incident, is a significant issue that has not been adequately addressed and therefore requires further research in its own right.

Intrusion response is defined as the process of counteracting the effects of an intrusion. It includes the series of actions taken by an Intrusion Detection System (IDS), following the detection of a security-related event. The justification for advancing the automated response capability of IDS is twofold: firstly, to reduce the significant overhead that manual response poses to the administration of increasingly large and complicated IT infrastructures, and secondly, to cope with the widespread use of automated scripts that can generate attacks of distributed nature.

In order to select appropriate responses, it is necessary to know more than just the type of incident that has occurred, or the basic security impact that has resulted. However, many current intrusion classification taxonomies provide little understanding beyond this level. As such, a new taxonomy has been developed as the basis for studying the issue of response, aiming to consider incidents and identify their different results in different contexts. It is intended that this taxonomy will give insight into the process of selecting appropriate responses and forming the basis of decision-making in an automated responder system (Furnell and Dowland 2000)

The discussion begins by summarizing previous work that has been conducted in relation to intrusion and attack taxonomies, before proceeding to present details of the new approach. The concept of the response-oriented taxonomy builds upon previous ideas, originally introduced by Furnell et al (2001).

CURRENT INTRUSION TAXONOMIES

Previous research has given rise to a number of intrusion taxonomies, each of which presents an alternative view of the situation. Brief summaries of a number of notable approaches are given below.

A common method of classifying security incidents is according to the impacts or outcomes resulting from their occurrence. This has led to a number of result-based taxonomies of incidents and attacks. In such approaches, all attacks are grouped into basic categories according to their result, aiming to give more insight into their severity. An example is a taxonomy devised by Cohen (1995) that ncludes result categories such as *Corruption, Leakage,* nd *Denial. Corruption* is defined as the unauthorised nodification of information, *leakage* is when information nds up where it should not be, and *denial* is when omputer or network services are not available for use. Another result-based taxonomy is specified by Russell nd Gangemi (1991), who define similar outcome ategories, but use a different set of terms (i.e. *secrecy* nd *confidentiality* instead of *leakage; accuracy, integrity,* nd *authenticity* instead of *corruption;* and *availability* nstead of *denial*).

Although result-based taxonomies can be useful in roviding a meaningful association between different ypes of attacks, the end result of an attack is not the only ignificant characteristic and thus it represents only one spect of the problem. In order to detect, respond, and pecify protection, it is necessary to have some lassification of the incidents that lead to the results. In his respect, there are also a number of prior works that an be considered.

Cheswick and Bellovin (1994) classify attacks into the even categories listed below:

- Stealing passwords - methods used to obtain other users' passwords
- Social engineering - talking your way into information that you should not have
- Bugs and backdoors - taking advantage of systems that do not meet their specifications, or replacing software with compromised versions
- Authentication failures2 - defeating of mechanisms used for authentication
- Protocol failures - protocols themselves are improperly designed or implemented
- Information leakage - using systems such as *finger* or the *DNS* to obtain information that is necessary to administrators and the proper operation of the network, but could also be used by attackers
- Denial-of-service - efforts to prevent users from being able to use their systems.

Although this approach provides a general overview, including the main categories of intrusions, it is not specified in any further detail, and thus is too general to provide any insight to the relationship among different classes of attacks or their different characteristics.

Neumann and Parker (1989) developed an intrusion taxonomy based on a large number of incidents reported to the Internet risks forum. The taxonomy classifies intrusions into nine categories, according to key elements that might indicate a particular type of incident. Table 1 below summarises the overall scheme.

NP 1 EXTERNAL MISUSE	Nontechnical, physically separate intrusions
NP 2 HARDWARE MISUSE	Passive or active hardware security problems
NP 3 MASQUERADING	Spoofs and Identity changes
NP 4 SUBSEQUENT MISUSE	Setting up intrusion via plants, bugs
NP 5 CONTROL BYPASS	Going around authorised protections/controls
NP 6 ACTIVE RESOURCE MISUSE	Unauthorised changing of resources
NP 7 PASSIVE RESOURCE MISUSE	Unauthorised reading of resources
NP 8 MISUSE VIA INACTION	Neglect of failure to protect a resource
NP 9 INDIRECT AID	Planning tools for misuse

Table 1: SRI Neumann-Parker taxonomy

An extension of the Neumann-Parker taxonomy was produced by Lindqvist and Jonsson (1997), which further refines security incidents into intrusions, attacks and breaches. It examines these issues from a system-owner perspective, based on a number of laboratory experiments. The results of these experiments indicated a need for further subdivision of the Neumann-Parker classes 5, 6 and 7, as shown in Table 2 below. Their work provides further insight into the process of spotting aspects of system elements that might indicate an intrusion.

Extended NP5 CONTROL BYPASS	Password attacks, spoofing privileged programs, utilizing weak authentication
Extended NP6 ACTIVE RESOURCE MISUSE	Exploitation of write permissions, resource exhaustion
Extended NP7 PASSIVE RESOURCE MISUSE	Manual browsing, automated browsing

Table 2: Lindqvist and Jonssen extension of the Neumann-Parker taxonomy

A final example is provided by Howard (1997), who follows a different approach by focusing on the process of an attack, rather than classification categories. Howard's taxonomy establishes a link through the different potential *attackers* (classified as hackers, spies, terrorists, corporate raiders, professional criminals and vandals) and the *tools* and *access methods* that they may utilise, leading to the *results* that enable the attackers to achieve their objectives. This taxonomy was based on the analysis of real incidents, as reported to the CERT/CC from 1989 to 1995, and thus represents a very valuable tool for systematically studying attacks. Having said this, it does not present a comprehensive top-level classification of intrusion incidents, or yield an appropriate classification that could be used to determine the required response – a criticism that could also be levelled at the other examples considered here.

Although most of the existing taxonomies succeed in contributing to the systematic study of intrusions, they are not immediately applicable to the domain of automated intrusion detection and response systems. From a detection perspective, it is clear that a number of the incident classifications identified (e.g. social engineering, physical tampering), and issues such as the objectives of attackers, could not be detected or determined by an automated system. In addition, they do not give any insight into the issue of response. A taxonomy that would serve this purpose ought to give consideration to the classification criteria, which will include aspects such as incident type, target, and/or potential impact. This will lead to indication of generic response categories, considering what can be done to halt an attack in progress, reduce its impact and/or prevent reoccurrence. The discussion of such a taxonomy is the focus of the next section.

A RESPONSE-ORIENTED TAXONOMY

The aim of the new taxonomy is to determine the effect an incident has on specific targets, and demonstrate how that may influence the response decision process. In order to demonstrate that concept a set of incidents have been used and are listed below:

1. Information gathering (Probe / Scan, Sniff)
2. Authentication failure (Masquerade / Spoof, Bypass)
3. Software compromise (Buffer Overflow, Flood / Denial of Service (DoS)
4. Malware (Trojan Horse, Virus / Worm)
5. Misuse (Unauthorised Alteration, Unauthorised Access)

As with the previous taxonomies, the selection of incidents is by no means exhaustive, but the five top-level categories aim to encompass the most significant set of incidents that affect current systems. Also, the description of the incidents used in the taxonomy aims to preserve a high level of abstraction, in order to include as many cases of incidents as possible. So, for example, although there are many different methods of launching Denial of Service attacks (such as SYN Flooding, SMURF attacks, Ping of Death, Trin00, and others), their ultimate effect on a system is similar, and it is this that will be the main determinant of the desired response(s). The five incident categories, and example incidents, are described more fully later in this section, following discussion of the other elements of the taxonomy.

Another important characteristic that can influence response is the *Target* of the intrusion, since the same incident can have different impacts upon different targets. The target groups considered in the new taxonomy are as follows:

- *External server:* Public-facing servers that are accessible from external networks and represent the public image of the host organization (e.g. web, email, DNS, FTP servers). Ideally, if configured correctly, external servers should not contain or facilitate access to confidential information, but ought to provide uninterrupted service to clients.
- *Internal server:* A server accessible only within the internal network of the organization (e.g. intranet web and file servers).
- *User workstation:* Computing units used by average users, likely to contain information specific to a particular user and their role within the organisation.
- *Network Component:* Networking equipment such as routers, switches, firewalls, which may be targeted as a means of accessing other systems or subverting operations.

This is by no means a detailed or exhaustive list, but it is sufficient to give a high level abstraction of the different elements that might be targeted in a typical organisation.

As well as the incident type and the target, the other significant characteristic that must be considered in order to select a response is the likely *result(s)* of an intrusion. However, this aspect cannot be represented in only one dimension, and the taxonomy presented here considers it to be comprised of *urgency*, *severity*, *impact(s)* and *potential incidents* arising from an incident.

The *Urgency* relates to the need for timely response, and partially reflects the speed of the attack. Since some attacks can evolve more rapidly than others, it is important to consider how much time is available to respond in each case. A Denial of Service attack, launched with the use of automated scripts is an example of a rapidly evolving attack, while sniffing traffic in a Local Area Network (LAN) allows a greater window of opportunity for response, as it is likely to evolve in a longer period of time. Another dimension of the result is the *Severity* of the intrusion, which relates to the magnitude or extent of the attack. The more severe an intrusion is, the sooner it needs to be contained, in order to eliminate its impacts and the threat introduced in the system. In the taxonomy, both urgency and severity are rated on a scale of Low, Medium, High for each incident / target combination.

Apart from the urgency and severity, another aspect of the result is the consideration of the *Impact(s)* of an intrusion upon a system. The *Impact(s)* relate(s) to the asset(s) of the system that have been compromised by the intrusion and may be observed and measured in relation to the *Confidentiality*, *Integrity* and / or the *Availability* of systems and data. Although in scenarios such as conventional risk analysis (Davey 1991) it is normal to rate these impacts on a sliding scale to indicate their severity, the taxonomy in the table that follows simply indicates whether there is a potential impact or not, as assignment of values would be too subjective.

The final element of the result relates to whether any further incidents are likely to be facilitated as a consequence of the initial attack. This is expressed in the taxonomy as *Potential Incidents*. For example, when sniffer software is used to capture network traffic, it is likely that the information obtained (e.g. user names and passwords) will enable attackers to log in as legitimate users at a later date and thus succeed in the masquerade. In other words, the potential incidents indicate the threat that has been introduced in the system after the occurrence of the original incident.

Having introduced the top-level elements of the taxonomy, the focus will now move to the incident categories identified earlier, as well as justifications to accompany the various ratings included in Table 3.

Information Gathering

The main characteristic of there intrusions is that they aim to collect information about a target and identify exploitable vulnerabilities. Although information gathering does not have significant impact upon a system, it carries the danger of the knowledge gained subsequently being used for launching other attacks with higher severity. Probe, Scan and Sniff are intrusions that fall into that category and will be described below.

Probe / Scan

Probe is used to access a target in order to determine its characteristics. Scan, on the contrary is used to access a set of targets in order to determine which of them have a specific characteristic. The characteristics in question aim to identify the architecture of targeted systems and networks, and usually relate to network configuration, as well as specific versions of services, operating systems and other types of software. The information obtained can subsequently enable the occurrence of incidents, such as spoofing, exploiting vulnerabilities and thus bypassing authentication, compromising software and introducing malware. The impacts relate to breach of confidentiality, as information is obtained without authorisation. Probing and especially scanning can also degrade availability, by producing large amounts of traffic when probing / scanning multiple targets. External servers as well as network components can be affected in this manner, as in both cases availability is highly important and it is those targets that are more likely to deal with that traffic.

The severity of scans / probes varies, depending on which target it is directed to. In the case of external servers and network components, which are genuinely subjected to unknown and thus untrustworthy users, they should be designed to be more tolerant with attacks of this nature. After all, within their normal activity they often provide the same nature of information anyway. Thus the severity of probing / scanning is not significant in those two cases. The urgency to respond is equally low, as apart from having low severity, probing / scanning is not likely to escalate rapidly. On the contrary, probing or scanning an internal server is not usual and thus it raises higher level of suspicion. Bearing in mind the importance of preserving confidentiality in internal servers, the level of high severity is more appropriate. The urgency to respond is medium, due to the high level of severity on one hand and its slow nature, in terms of escalating on the other. As for user workstations, although probing / scanning a user workstation is even more rare and thus raises higher level of suspicion, its impact is not as severe, as the threat to confidentiality in this case is significantly lower. Thus the severity can be regarded as 'medium'. However, the occurrence of such an incident could mean prior breach of another target (e.g. DNS server), and thus a medium level of urgency to respond is considered appropriate.

Sniff

Sniffing consists of the interception of traffic while it travels across the network. It is achieved with the use of software tools that can capture network packets either locally or remotely. The sort of information obtained with sniffing could be anything that travels across the network, such as user name and password combinations, data files, and system or network information. After obtaining information with sniffers, the potential incidents likely to follow can mainly be masquerading, bypassing, and software compromise.

The impacts of sniffing mainly involve loss of confidentiality, however its severity and urgency depend on the type of targets subjected to it. In external servers the severity is low, since again the nature of information disclosed cannot be significant enough to raise the level of severity. Similarly with probing / scanning, the need for timely response is low, since the severity of the incident and the chance of escalating are low. In the case of internal servers, the severity is again high, however the need to respond is high as well, since the nature of information that can be disclosed in this case is more significant and thus requires a more urgent issue of response. As for user workstations, the nature of information exposed is not significant enough to increase the level of severity and urgency, so as in the case of probing / scanning, both are considered as medium.

INCIDENT	TARGET	RESULT					
		URGENCY	SEVERITY	IMPACT			POTENTIAL INCIDENTS
				C	I	A	
1.Information gathering							
Probe / Scan	External server	Low	Low	✓		✓	Spoof, Bypass, S/w compromise, Malware
	Internal server	Medium	High	✓			
	User workstation	Medium	Medium	✓			
	Net. component	Low	Low	✓		✓	
Sniff	External server	Low	Low	✓			Masquerade, Bypass, S/w compromise
	Internal server	High	High	✓			
	User workstation	Medium	Medium	✓			
	Net.component	Medium	Medium	✓			
2. Authentication failure							
Masquerade / Spoof	External server	High	High	✓		✓	Misuse, Malware, Software compromise
	Internal server	High	High	✓			
	User workstation	Medium	Medium	✓			
	Net. component	High	High	✓		✓	
Bypass	External server	High	Medium	✓			Misuse, Malware
	Internal server	High	High	✓			
	User workstation	High	Medium	✓			
	Net. component	High	Medium	✓			
3. Software Compromise							
Buffer Overflow	External server	High	High		✓	✓	Bypass, DoS, Misuse, Malware
	Internal server	High	High		✓	✓	
	User workstation	High	Medium		✓	✓	
	Net. component	High	Medium		✓	✓	
Flood / DoS	External server	High	High			✓	Spoof
	Internal server	High	High			✓	
	User workstation	Medium	Medium			✓	
	Net. component	High	High			✓	
4. Malware							
Trojan Horse	External server	High	High	✓	✓	✓	Bypass, Misuse, Malware, S/w compr., Info. gathering
	Internal server	High	High	✓	✓	✓	
	User workstation	High	High	✓	✓	✓	
	Net. component	High	High	✓	✓	✓	
Virus / Worm	External server	High	High	✓	✓	✓	Misuse, Malware, S/w compr., Info. gathering
	Internal server	High	High	✓	✓	✓	
	User workstation	High	High	✓	✓	✓	
	Net. component	High	High	✓	✓	✓	
5. Misuse							
Unauthorised Alteration	External server	High	High		✓	✓	Malware
	Internal server	High	High		✓	✓	
	User workstation	High	Medium		✓	✓	
	Net. component	High	High		✓	✓	
Unauthorised Access	External server	High	Low	✓			Malware, Unauthorised Alteration
	Internal server	High	High	✓			
	User workstation	High	Medium	✓			
	Net. component	High	Low	✓			

Table 3: Response–oriented Intrusion Taxonomy

Finally, in the case of network components, the severity of sniffing is medium, since the nature of information exposed in this case (e.g. Access Control Lists, administrator user account details) is significant enough to raise the level of severity. The urgency to respond is also medium, since network components represent single points of failure and a possible compromise could affect multiple hosts.

Authentication failure

Users and processes need to identify and authenticate themselves quite often in order to obtain specific access privileges. As a result, defeating the authentication process is very common objective for attackers, and can be summarised in three main ways, namely Masquerading, Spoofing and Bypassing.

Masquerade / Spoof
Masquerade is the action in which valid identification and verification information that belongs to legitimate users is obtained and used by an impostor. For example, an attacker might use a sniffer to capture user name, password and IP address combinations that are sent across the network, and then use this information to log into accounts that belong to other users. Spoofing, by contrast, involves the provision of false information. In network communications, each packet of information traveling on a network contains source and destination addresses either in the form of MAC, IP addresses, TCP connection IDs, or port numbers. Supplying accurate information is often assumed, however it is possible that incorrect information is entered into these communications, in order to accept an impostor address as original and either trick other machines into sending it data or to allow it to receive and alter data. Examples include IP spoofing, email spoofing and DNS spoofing.

Masquerading and spoofing are mainly a threat to the confidentiality of systems, since they most often provide unauthorised increased access to attackers. However, in the case of external servers and network components, it is possible to cause loss of availability as well, if used as a technique to enable the occurrence of DoS attacks. The potential incidents that can follow masquerading and spoofing are obviously misuse (unauthorised access and alteration of information), malware (introduction of Trojan horses, viruses / worms) and software compromise (Buffer overflow, DoS).

The severity of masquerading and spoofing is considered high in external servers, as it may result in loss of availability. The urgency to respond is high as well, since IP spoofing can very soon escalate to a DoS incident. However, even in the case of masquerading, once unauthorised access is achieved to external servers, it is possible to alter information that can harm the public image of the organisation and thus cause further embarrassment and disruption of operation. In the case of internal servers, even if services are not accessed externally, the danger of disclosing confidential information is considerably high, resulting in severe embarrassment to the organisation, and disruption of its operation. So, the level of severity and the urgency to respond in this case are high as well. As for user workstations, the severity is less significant, as in many cases the nature of information or access level obtained will not pose a great level of threat to the system (although some users will always be exceptions). The level of urgency is medium as well, since the workstation is probably used as a step to achieve increased access into a more significant component of the system (either internal or external server). Obtaining unauthorised access in network components, as well as making them unavailable by achieving DoS attacks is highly severe, as it can affect multiple hosts or even the entire internal network, depending on the scale of the problem. The urgency to respond is thus high as well.

Bypass
Bypass is an action taken to avoid the authentication process by using an alternative method to access a target. For example, some operating systems have vulnerabilities that could be exploited by an attacker to gain privileges without actually logging into any privileged account. Bypass is usually a result of software compromise (e.g. buffer overlow) or malware (e.g. if a trojan horse is used instead of the original authentication process). The issue is again a threat to confidentiality, as increased unauthorised access is achieved. The potential incidents that can follow are misuse (unauthorised access and alteration of information) and malware.

The severity is medium in the case of external servers, since their availability is not threatened directly. However a rapid response is needed to avoid further escalation of the incident, so the urgency in that case is high. In internal servers both severity and urgency are high, as the direct threat is higher, so is the need to avoid escalation of the incident. Although the severity in the case of user workstations is lower, and thus can be considered as medium, the need to respond is equally high, since bypassing authentication is an indication of an already compromised system, so further action should be taken as soon as possible. Finally, bypassing authentication in network components is of medium severity, since the threat to confidentiality is not as severe as in the case of internal servers, but again the need to respond and eliminate any chances of escalating the problem is high.

Software compromise

Intrusions that involve the exploitation of software vulnerabilities fall into this category. There are three main categories of vulnerabilities within a system, namely design, implementation or configuration vulnerabilities (Howard 1997). The main categories of intrusions that fall into this category are Buffer Overflow and Denial of Service; they are presented below.

Buffer Overflow
Buffer overflow is a result of deficient software implementation that allows the assignment of data in a buffer without checking in advance if its size is sufficient to 'host' that data. So in the case of someone sending larger amounts of data, the targeted system will allow the input of data in the buffer anyway, with the result of either crashing the system or overwriting part of memory adjacent to the buffer. As a result of the latter, unauthorised access could be obtained by modifying the flow of program execution, and allowing the execution of arbitrary code with the same access rights granted to the compromised program (Aleph1 1996).

Such incidents can compromise the integrity and availability of the targeted system, and can lead to further incidents such as bypassing authentication, denial of service, misuse or execution of malware. In all cases, the amount of time elapsing before that happens is usually small, as in many cases it even happens almost simultaneously.

Buffer Overflows are more commonly exploited in server software (web, ftp, email, file) since they are easily accessible from external sites and often run under root/administrator privileges. Thus high potential severity can exist for external servers, as well as internal (intranet) servers in some organisations. The urgency to respond is high as well, since apart from the significant severity of the incident, the likelihood of escalation is significant as well, so an urgent response is needed.

In the case of user workstations the severity is medium, since the chances of being subjected to attacks of this nature is less substantial. Also, even if targeted (e.g. server software is running, probably by default) the number of hosts affected are limited (probably only one), so the scale of the problem is less significant. However, the urgency to respond is still high, in order to avoid execution of malware or further compromise of other systems.

The chance of exploiting buffer overflows in network components is even less significant, but the potential impacts of doing so are more substantial than in the case of workstations, since a greater number of hosts can be affected. Thus the severity of buffer overflow is medium in this case. The urgency to respond is again high, for the same reason.

Flood / Denial of Service
Denial of Service (DoS) attacks aim to overload (flood) the capacity of a target by accessing it repeatedly. The result of such action is to make the target unable to respond to any other events / requests and thus become inaccessible to legitimate clients. Subsequent occurrences could include another party assuming the role of the target, resulting in spoofing.

The impact of Denial of Service attacks clearly relates to the availability of the targets. Since these attacks are most often conducted with the use of automated scripts, the need to respond immediately is crucial in most cases. In the case of an external server, the severity is likely to be high, given that a site may represent a public interface of the organization. Inaccessibility could result in embarrassment and loss of custom. The urgency to respond is also high, since usually the time available to prevent either the occurrence of the incident, or subsequent escalation, is very limited. Although DoS to internal servers and network components does not risk causing embarrassment to the organisation, their failure to provide services could have impact on multiple hosts, or even the entire internal network of the organisation, so the severity is also high, as is the urgency to respond. In the case of user workstations, the likelihood of being subjected to a DoS attack is rather small, simply because the impact of doing so is not as significant. User workstations are mostly used as (potentially unwitting) tools to conduct DoS attacks in order to achieve maximum level of effectiveness, but are not the targets. However, it is possible, and it can result in either degradation of performance, or total loss of legitimate usability. Thus the severity in that case is medium. The urgency to respond is medium as well, as the impacts of the attack are of medium severity and the time available to encounter the attack or avoid escalation is usually more.

Malware

Malicious software, also known as malware, characterises the classes of intrusions that are conducted under complete software control. Intrusions falling into this category differentiate from automated software tools used to launch other classes of attacks (e.g. DoS attacks), in the sense that humans are not involved in the escalation of malware attacks; after the initial human involvement to begin the distribution of malware, individual attacks can subsequently occur without the need for the instigator's further involvement. Thus malware can constitute an attack in its own right. There are three main types of malware, namely Trojan horses, viruses and worms and will be discussed below.

The impacts of malware can differ significantly from case to case, since the code in the payload can do nearly everything that is feasible under software control. For example, it is possible to initiate posting of legitimate users' working documents to all the members of his/her address book, resulting to breach of confidentiality (SARC 1999). Alternatively, it is possible to delete or modify files in the system, achieving a breach of integrity. Finally system or network resources can be consumed at the execution of the payload, resulting to either degradation of performance or entire inaccessibility of targets for legitimate use.

The potential incidents that can follow the execution of malware can also be nearly anything. Misuse, other forms of malware, software compromise and information gathering are examples of potential results of malware. Thus the severity of malware varies according to the specific incidents. However, if considering the execution of malware in general, the severity is high in all types of targets, since such a great variety of functionality can potentially be included in the payload. In addition, the risk of spreading to additional targets is extremely high, so the urgency to respond and contain the execution of malware is high as well in all cases.

Misuse

Misuse relates to unauthorised or unacceptable use of system resources. In this sense, it is a quite general term that can actually include all the incidents described so far, since all of them are somehow a form of misusing system resources. However, incidents falling into this category mainly take place after unauthorised access has been obtained in a target and include cases that mainly involve misuse of files and data within a system. It is important to mention at this point that the occurrence of incidents from this category indicates that the targeted system may have already been in a compromised state, unless the activity is being perpetrated by a legitimate user. Hence any response issued might be affected by this factor as well.

Unauthorised alteration

Unauthorised alteration includes actions such as creating, modifying, deleting system or data files. This will affect the integrity and / or availability of resources and represents an important issue that needs to be addressed.

The severity in the case of external servers is high, as information or services might be altered in such a way as to cause embarrassment to an organisation and further disruption to its normal operation. For example, web site defacements (Alldas.de 2001) represent a highly important incident that can immediately attract the interest of media and put the organisation into a difficult situation. Also the modification of information or services could potentially mislead or cheat customers, and result in making the organisation liable for those actions. Although the urgency to respond in such case is high, the feasibility of doing so might be another issue. Certainly the current state of the system needs to be considered in order to determine the effectiveness or selection of an appropriate response.

Unauthorised alteration is highly severe in the case of internal servers and network components as well, since it can result in misleading internal users to make decisions based on inaccurate information or disrupting their operation. Even if the likelihood for rapid escalation of the incident is very small, the need for timely response is high again, since the severity of the incident is so significant.

Finally in the case of user workstations, the importance of the target is typically lower, as it can affect only a limited number of users. The severity is therefore medium. Still, the urgency to respond is high, mainly because the current state of the targeted system should be assessed and any potential risks minimised.

Unauthorised Access

Unauthorised access includes actions that involve disclosure of information to unauthorised parties. As a result of their occurrence, incidents such as unauthorised alteration or execution of malware might follow. Thus the severity of unauthorised access can vary according to the target (and whether confidentiality is at high risk) but the urgency to respond in all cases should be high. That is to firstly assess the current state of the system and prevent further escalation of the incident and occurrence of unauthorised alteration or execution of malware as well.

When external servers or network components are subjected to unauthorised access, the severity is low, since no confidential information should be at risk and no modification has taken place. On the other hand the current state of the system is unknown and needs to be assessed. By contrast, unauthorised access to internal servers has high severity, because there is more important information available for attackers. In the case of user workstations the severity is medium, as there is risk to confidentiality, but it is less substantial.

CONCLUSIONS

In this taxonomy, several incidents have been considered, aiming to illustrate the effect of different types of targets on the results of an intrusion. The ultimate intention is to give insight into the main intrusion characteristics that can influence intrusion response, and subsequently lead to the indication of generic classes of response. Although the response-oriented taxonomy is quite generic and cannot depict the complexity of the response decision process, it can still serve as a basic tool that will enable the research to progress towards that direction. After looking into the results of different intrusions on various targets, it seems that intrusions directed towards internal servers always have the most significant results, mainly due to their importance in the operation of an organisation. By contrast, user workstations have the least significant results, as their role within the organisation is less important and the consequences after the occurrence of an intrusion can more easily be addressed. Finally, network components and external servers seem to depend on the type of intrusion to a greater extent, as some classes of intrusions have more significant results than others.

In terms of response and how different intrusion characteristics can influence the response process, it can be argued that the more severe an intrusion is, the more important it is for the response to focus on the prevention of its occurrence, or its containment. In classes of intrusions with low or medium severity and high urgency, the risk for rapid escalation is significant, and so the response process should focus on the prevention of further escalation (prevent the occurrence of potential incidents). Finally, the severity and urgency can affect the transparency of the initiated response. It seems that there should be a trade-off between them, as the more severe the intrusions, the less transparent responses can apply.

It should be noted that there are several limitations in this taxonomy. For example, apart from the type of target, the number of systems targeted could also be considered, as the scale of an incident will certainly influence its severity. For example, a virus that infects a small number of user workstations is not as severe as one that infects all of them. However, the omission of this factor does not prevent the taxonomy from fulfilling its objective of demonstrating that the same category of incident can demand different responses in different contexts.

As regards the responses themselves, it may appear curious that they have been omitted from the taxonomy presented here. The basic reason is that the taxonomy is intended to provide the foundation for an automated decision mechanism within a software agent. The specific response options available could vary depending upon the environment in which the agent is deployed, and thus the classification taxonomy is independent of any particular mapping. In the context of such an agent, the decision-making process could also be more complex. Although incident and target related characteristics are the main determinant of the likely result of the incident, various other contextual factors could be measured when an incident is detected in order to better inform the response decision process. For example, the account in use, the current alert level of the IDS, and the nature of any responses already issued could all influence the choice of response that is likely to be the most effective. Further consideration of this issue is presented in (Papadaki et al. 2002), and the issue represents the focus of ongoing research by the authors.

REFERENCES

Aleph1 (1996), "Smashing The Stack For Fun And Profit", Phrack online journal, vol. 7, issue 49, 8 November 1996.

Alldas.de (2001), "Defacement Archive", http://defaced.alldas.de/, 26 October 2001.

Cheswick W.R., and Bellovin S.M. (1994), 'Firewalls and Internet Security: Repelling the Wily Hacker', Addison-Wesley Publishing Company, 1994.

Cohen F.B. (1995), *Protection and Security on the Information Superhighway*, John Wiley & Sons.

Davey J. (1991), "The CCTA risk analysis and management methodology (CRAMM)", Current Perspectives in Healthcare Computing, pp. 360 – 365.

Furnell S.M. and Dowland P.S. (2000), "A conceptual architecture for real-time intrusion monitoring", *Information Management & Computer Security*, Vol. 8, No. 2, pp65-74.

Furnell, S.M, Magklaras, G.B, Papadaki, M. and Dowland, P.S. (2001), "A Generic Taxonomy for Intrusion Specification and Response", in *Proceedings of Euromedia 2001*, Valencia, Spain, 18-20 April 2001: 125-131

Howard J.D. (1997), PhD thesis 'An Analysis of Security Incidents on the Internet 1989 - 1995', Carnegie Melon University, 7 April 1997, http://www.cert.org/nav/reports.html

Lindqvist U., and Jonsson E. (1997), "How to Systematically Classify Computer Security Intrusions", in Proceedings of the 1997 IEEE Symposium on Security and Privacy, May 4-7, 1997, IEEE Computer Society Press.

Neumann P.G., and Parker D.B. (1989), "A summary of computer misuse techniques", in Proceedings of the 12th National Computer Security Conference, Balitimore, USA, 10-13 Oct 1989, pp. 396-407.

Papadaki, M., Furnell, S.M., Lee, S.J., Lines, B.M. and Reynolds, P.L. (2002), "Enhancing response in intrusion detection systems", submitted to *Journal of Information Warfare*.

Russell D. and Gangemi G. T. (1991), "Computer Security Basics", O'Reilly & Associates, Inc., Sebastopol, CA, 1991.

SARC. 1999. "W97.Melissa.A virus overview". Symantec AntiVirus Research Center. http://service1.symantec.com/sarc/sarc.nsf/html/W97.Melissa.A.htm

ACKNOWLEDGEMENTS

The research presented in this paper has been supported by funding from the State Scholarships Foundation (SSF) of Greece.

17

KEYSTROKE ANALYSIS AS A METHOD OF ADVANCED USER AUTHENTICATION AND RESPONSE

P.S.DOWLAND, S.M.FURNELL and M.PAPADAKI

nrg@plymouth.ac.uk
Network Research Group
Department of Communication and Electronic Engineering
University of Plymouth
Drake Circus
PLYMOUTH
PL4 8AA
United Kingdom
Tel: +44 1752-233521 *Fax: +44 1752-233520*

Key words: Keystroke Analysis, User Authentication, Biometrics, Intrusion Response.

Abstract: There has been significant interest in the area of keystroke analysis to support the authentication of users, and previous research has identified three discrete methods of application; static, periodic dynamic and continuous dynamic analysis. This paper summarises the approaches and metrics arising from previous work, and then proceeds to introduce a new variation, based upon application-specific keystroke analysis. The discussion also considers the use of keystroke analysis as a progressive, escalating response measure in the context of a comprehensive user authentication and supervision system, presenting an example of how this could be realised in practice.

1. INTRODUCTION

The issue of user authentication in IT systems has long been recognised as a potential vulnerability, with the majority of current systems relying upon password methods. Such methods have been repeatedly proven to be open to compromise, and can also be considered problematic in the sense that they typically only serve to facilitate a one-off authentication judgement at the start of a session. A number of previous works [1, 2, 3] have consequently discussed the need for some form of monitoring to continuously (or periodically) authenticate the user in a non-intrusive manner. Although such monitoring is technically feasible, there are significant issues to be considered in selecting appropriate attributes to assess. This is particularly important, as continuous monitoring must be transparent to the end user in order to minimise any perceived inconvenience (with the exception of appropriate challenges in the event of suspected impostor activity).

A number of studies have considered the application of keystroke analysis to the problem of inadequate user authentication in modern IT system using static [4, 5, 6] and dynamic [7, 8] implementations. While these studies have evaluated the effectiveness of the proposed solutions, none have considered the implementation and necessary supporting application framework to effectively use keystroke analysis as a viable authentication and supervision mechanism.

This paper summarises the potential approaches to keystroke analysis, and presents details of a new method based on application-specific user profiling. It then proceeds to consider how keystroke analysis may be utilised as part of an intrusion response framework.

2. KEYSTROKE ANALYSIS OVERVIEW

Previous studies have identified a selection of data acquisition techniques and typing metrics upon which keystroke analysis can be based. The following section summarises the basic methods and metrics that can be used.

- **Static at login** - Static keystroke analysis authenticates a typing pattern based on a known keyword, phrase or some other predetermined text. The captured typing pattern is then compared against a profile previously recorded during system enrolment. Static keystroke analysis is generally considered to be an initial login enhancement as it can supplement the traditional username/password login prompt, by checking the digraph latencies of the username and/or password components (i.e. authenticating the user on the basis of both *what* they typed and *how* they typed it).

- **Periodic dynamic** - Dynamic keystroke analysis authenticates a user on the basis of their typing during a logged in session. The captured session data is compared to an archived user profile to determine deviations. In a periodic configuration, the authentication judgement can be intermittent; either as part of a timed supervision, or, in response to a suspicious event or trigger. This method provides distinct advantages over the static approach. Firstly, it is not dependent on the entry of specific text, and is able to perform authentication on the basis of any input. Another factor is the availability of data; in static keystroke analysis, the range of digraphs and frequency of their occurrence is likely to be significantly limited compared with a dynamic approach. Even an inexperienced typist is likely to produce sufficient digraph pairs to allow an authentication judgement to be derived. This is an important factor as it is necessary to have a statistically significant volume of keystroke data in order to generate a user profile.

- **Continuous dynamic** - Continuous keystroke analysis extends the data capturing to the entire duration of the logged in session. The continuous nature of the user monitoring offers significantly more data upon which to base the authentication judgement. With this method it is possible that an impostor may be detected earlier in the session than under a periodically monitored implementation. On the downside, however, the additional processing required will add to the computational overhead of the supervision system.

- **Keyword-specific** - Keyword-specific keystroke analysis extends the continuous or periodic monitoring to consider the metrics related to specific keywords. This could be an extra measure incorporated into a monitoring system to detect potential misuse of sensitive commands. For example, under a DOS/Windows environment it may be appropriate to monitor the keystroke metrics of a user attempting to execute the FORMAT or DELETE commands. This could represent a significant enhancement, as a command with a high misuse consequence (e.g. DEL *.*) is unlikely to cause sufficient profile deviation when observed from a system-wide context, due to the limited selection of digraphs. By contrast, static analysis could be applied to specific keywords to obtain a higher confidence judgement.

- **Application-specific** - Application-specific keystroke analysis further extends the continuous or periodic monitoring. Using this technique, it may be possible to develop separate keystroke profiles for distinct applications. For example, a user may be profiled separately for their word processing application and their email client. The potential of this new technique is discussed in more detail in section 3.

In addition to a range of implementation scenarios, there are also a variety of possible keystroke metrics that can be profiled as the basis for subsequent comparison:

- **Digraph latency** - Digraph latency is the metric that has traditionally been used for previous studies, and typically measures the delay between the key-up and the subsequent key-down events, which are produced during normal typing (e.g. T-H). In most cases, some form of low and high pass filter is applied to remove extraneous data from the session data.

- **Trigraph latency** - Trigraph latency extends the previous metric to consider the timing for three successive keystrokes (e.g. T-H-E).

- **Keyword latency** - Keyword latencies consider the overall latency for a complete word or may consider the unique combinations of digraph/trigraphs in a word-specific context.

- **Mean error rate** - The mean error rate can be used to provide an indication of the competence of the user during normal typing. Whilst this may not be user specific, it may be possible to classify users into a generic category, according to their typing ability, which can then be used as an additional authentication method.

- **Mean typing rate** - A final metric is that of the mean typing rate. As with the mean error rate, individuals can be classified according to their typing ability and hence evaluated based on their average typing speed.

While the final two metrics indicated above are unlikely to provide a suitably fine-grained classification of users for direct authentication judgements, they may be used to provide a more generic set of user categories that can contribute to a combined measure.

It should be noted that all of the above techniques and metrics can be implemented on a standard PC platform, without the need for special hardware.

3. EXPERIMENTAL DYNAMIC KEYSTROKE ANALYSIS

The idea of using keyboard characteristics for authentication is not unique, and there have been a number of previous published studies in the area. To date, however, virtually all published studies have focussed upon static or context-independent dynamic analysis, using the inter-keystroke latency timing method. From the earliest studies in 1980 [9], the focus has been on the analysis of digraph latencies. Later studies [6, 8] further enhanced the work, identifying additional statistical analysis methods that provided more reliable results.

In [7], the concept of dynamic keystroke analysis was first proposed, with the introduction of a reference profile that could be used to monitor a live user session. Brown and Rogers [5] also explored the idea of dynamic analysis, presenting preliminary results.

A summary of some of the main results from studies to date is presented in *Table 1* below, which illustrates the effectiveness observed (in terms of false acceptance and false rejection errors), as well as the type of keystroke analysis technique employed (digraph/trigraph etc.) and the analysis approach taken (statistical/neural network etc.).

Table 1: Previous keystroke analysis studies

Authors	Method	%FAR	% FRR
Umphress & Williams (1985) [10]	Digraph Statistical	6%	12%
Legget & Williams (1988) [11]	Digraph Statistical	5%	5.5%
Joyce & Gupta (1990) [6]	Digraph Statistical	0.25%	16.67%
Bleha et al. (1990) [12]	Digraph Statistical	2.8%	8.1%
Legget et al. (1991) [7] ^{1}Static, ^{2}Dynamic	Digraph Statistical	5% ^{1} 12.8% ^{2}	5.5% ^{1} 11.1% ^{2}
Brown & Rogers (1993) [5] ^{1}Group 1, ^{2}Group 2	Digraph Combined Neural Network & Statistical	0%	4.2% ^{1} 11.5% ^{2}
Napier et al. (1995) [13]	Digraph Statistical	29.5% / 3.8%	
Mahar et al. (1995) [8]	Digraph Statistical	35% / 17.6%	
Furnell et al. (1996) [14] ^{1}Static, ^{2}Dynamic	Digraph Neural Network ^{1}, Statistical ^{2}	8% ^{1} 15% ^{2}	7% ^{1} 0% ^{2}

A further variation in the data analysis can be introduced through the consideration of application specific keystroke profiles. If we accept from previous work that individual users have a distinct typing pattern, it can be hypothesised that an individual's typing pattern may also vary depending upon the application in use. For example, a user participating in a chat session may type in a fairly relaxed style, while the same user may type in an significantly different way when producing a document. It should also be noted that certain categories of user might use the numeric keypad for large quantities of data entry. Under these circumstances the volume and diversity of the keystroke digraphs will vary tremendously when compared to the more usual alphanumeric typing encountered with most user profiles. Previous research has been carried out in this area [15], which has shown that analysis of numeric keystrokes can provide a viable authentication measure. This is an area receiving on-going attention through a separate research project at the authors' institution.

In [16] the authors described a trial in which keystroke data, obtained within Microsoft Windows NT, was evaluated across all applications. While the results from this trial were encouraging, the quantity of data collected was insufficient to make a true, statistically valid, conclusion. Instead it was determined that further trials were necessary. Following the first trial, the authors conducted a second round of monitoring in which eight test subjects were profiled. Over a period of 3 months, a total of 760,000 digraph samples were captured and stored for analysis. In this case, however, the analysis was conducted with a view to determining viability of application-specific keystroke profiling. To this end, it was necessary to identify a series of applications for profiling, with the selection criteria being those for which sufficient keystroke data had been logged during the sampling period. A review of the keystroke data revealed that the applications satisfying this requirement were Microsoft MSN Messenger, Internet Explorer, Word and PowerPoint. While the authors considered that a numerically intensive application such as Excel would have provided an interesting candidate, insufficient keystrokes were captured to enable the creation of a profile. Additionally, of the eight users sampled during the trial, only five produced sufficient data to analyse from all of the aforementioned applications. Although the resulting sample group was very small, it was sufficient to yield interesting results in relation to an initial assessment of application-specific profiling.

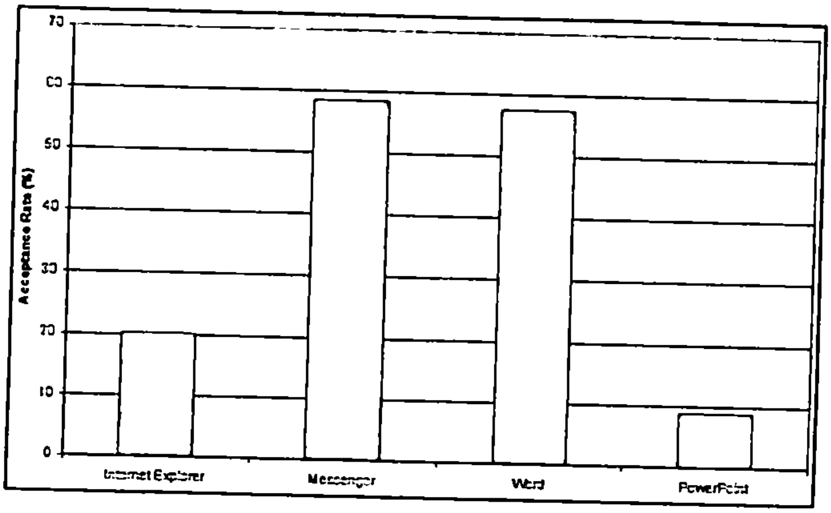


Figure 1: Acceptance Rate for application specific keystroke data compared against a system-wide context user profile

In *Figure 1* above, a single user's application-specific keystroke data is compared against the reference profile from the same user. The reference profile was based on all keystroke data acquired from all applications. Although the figure does not show distinct differences in all cases, there is a clear distinction between all applications apart from Messenger and Word. This can be explained when the nature of these applications is considered. Messenger and Word are both significantly textual in their usage, and users will typically type within Messenger and/or Word for considerable periods of time. In contrast, while Internet Explorer and PowerPoint sessions may both involve significant elements of keyboard activity, the typing is more likely to occur in sporadic bursts. As such, any dynamic that emerges is likely to be markedly different to that which would emerge in applications where more sustained typing is the norm. Considering the information portrayed above, the creation of application specific profiles would be likely to increase the acceptance rates observed.

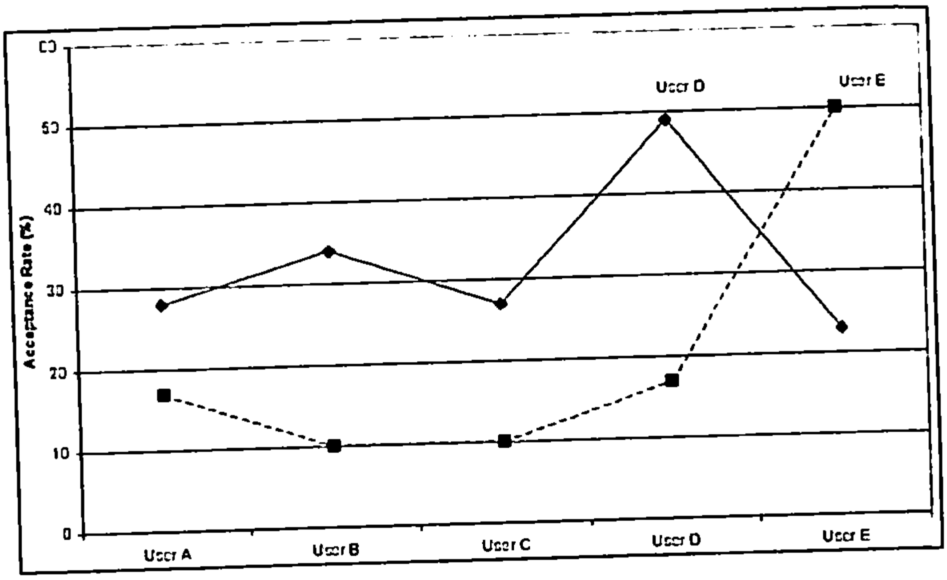


Figure 2: Acceptance Rate for two user profiles

In *Figure 2* above, a specific users' profile (users D and E when using Internet Explorer) is examined, showing there is a clear difference between other users' keystroke data (impostors) with appropriate peaks in acceptance rate for the valid users.

While the results shown do not indicate a suitably discriminative metric upon which to base a satisfactory authentication judgement, they do show a level of correlation between a user's typing pattern in an application-specific context. These preliminary results show that further work is needed to investigate the use of application-specific keystroke analysis.

4. AN ESCALATING RESPONSE FRAMEWORK USING KEYSTROKE ANALYSIS

The earlier discussion summarised the different potential implementations of keystroke analysis, and explained the operational differences between the approaches. It is possible to integrate these analysis approaches into an overall user authentication and supervision framework, with the varying techniques being invoked as responses to anomalies detected at earlier stages. A possible example of this is illustrated in *Figure 3*, which shows how the five variations discussed earlier can be incorporated within a four-level response framework. It should be noted that this is by no means the only method by which the techniques could be combined, and specific implementations could vary depending upon rule sets for a particular user, class of users, or general organisational security policy.

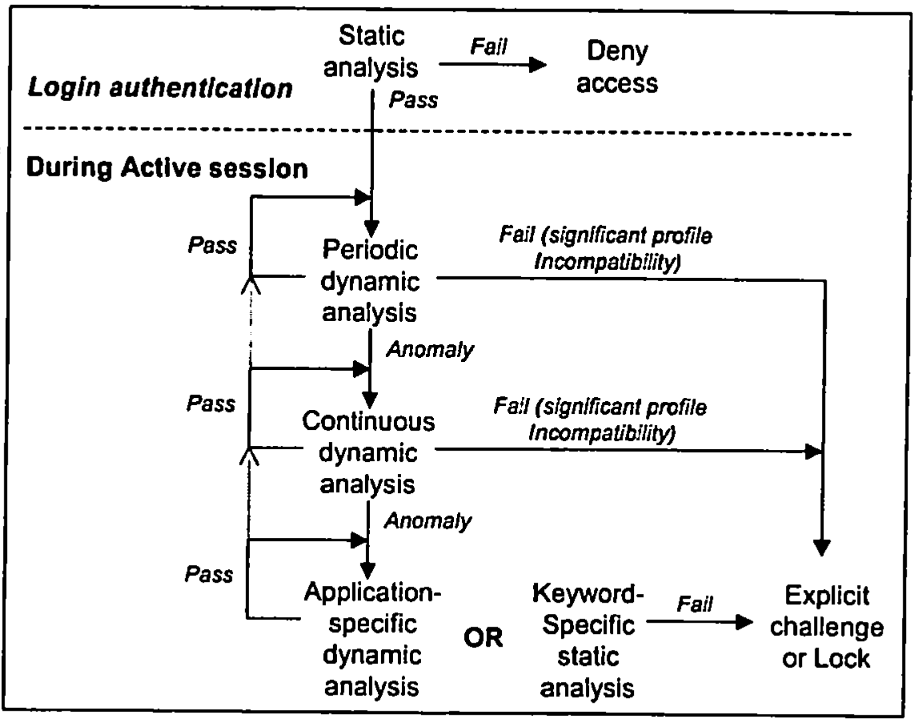

Login authentication
Static analysis
Fail
Deny access
Pass
During Active session
Pass
Periodic dynamic analysis
Fail (significant profile incompatibility)
Anomaly
Pass
Continuous dynamic analysis
Fail (significant profile incompatibility)
Anomaly
Pass
Application-specific dynamic analysis
OR
Keyword-Specific static analysis
Fail
Explicit challenge or Lock

Figure 3: Response framework using keystroke analysis

A suitable architecture for achieving such an approach is offered by the Intrusion Monitoring System (IMS) [17]. This proposes an architecture for real-time user authentication and misuse detection, based upon a monitoring *Host* that has the responsibility for supervising a number of *Client* systems (e.g. in the form of end user PCs or workstations). Key elements of the architecture, from the perspective discussed in this paper are the *collector* (which obtains the keystroke data from the individual client systems), the *anomaly detector* (which performs the actual keystroke analysis and profile comparison, maintaining a consequent alert status metric), and the *responder* (which is responsible for initiating the different keystroke analysis approaches in response to increases in the alert status and other contextual factors). Assuming such a monitoring context, the text below describes how the response process in *Figure 3* would proceed.

Initial authentication may occur using a standard username/password pair, but supplemented by the use of static keystroke analysis to assess how the information is entered. If the user fails to authenticate at this stage (e.g. after being permitted three attempts to enter the details), then the most appropriate response is to deny access (if the correct password is provided, but the keystroke analysis aspect fails, then an alternative option could be to allow the login to proceed, but to begin the session with a higher level of subsequent monitoring – e.g. continuous rather than periodic assessment). If this login authentication is successful, the user will proceed to a logged in session, during which dynamic keystroke analysis could be applied on a periodic basis (in order to minimise the associated processing overhead in the initial instance). Assuming no anomalies, this could simply continue throughout a logged in session. If a departure from the typing profile is noted during the monitoring period, however, there would be two options for response. If the keystroke data exhibits a significant incompatibility, then a high confidence of impostor action could be assumed and the responder could proceed directly to some form of explicit action (e.g. interrupting the user session by issuing a challenge or suspending their activity pending an administrator intervention). In cases where the profile incompatibility is not conclusive, the responder could initiate an increase in the monitoring resolution – firstly to invoke continuous dynamic analysis, and then beyond this to invoke either application or keyword-specific methods. The choice in the latter case would depend upon the context of the current user's activity. For example, if they were word-processing, then application-specific dynamic analysis would potentially give a more accurate assessment of identity. If, by contrast, they were operating at a command line level, then it could be considered more appropriate to invoke keyword-specific static analysis, looking for instances of particularly sensitive commands such as 'format' or 'erase'. Profile incompatibility at this final stage would automatically result in more explicit response action.

In cases where the responder agent has initiated a more detailed level (e.g. from periodic to continuous, or from continuous to application-specific), then the monitoring would continue at this level for a period of time, in order to ensure that profile incompatibilities were no longer observed. A suitable trigger (e.g. the entry of a certain number of further keystrokes without significant profile departure) would be used to reduce the alert status of the monitoring system, and thereby allow the responder agent to re-invoke a lesser level of analysis (this is indicated by the dotted arrow lines in the figure).

The combination of mechanisms in this manner allows a system to provide a standard, and hence acceptable, user login for the initial authentication, while also providing enhanced user supervision for the duration of the users' session. Such a system should, in theory, ensure transparent operation to legitimate users. It should also be noted that, in a practical context, keystroke analysis may not be the only technique involved, and other metrics relating to user activity and behaviour might also be considered by the *anomaly detector*, and thereby used to inform the *responder* agent.

5. CONCLUSIONS

This paper has considered the significant variety of implementation methods and metrics that can be associated with keystroke analysis. The new concept of application-specific analysis has been introduced, along with initial experimental findings that support the feasibility of the approach. The preliminary results suggest that the technique is worthy of further investigation.

The discussion has also considered the application of keystroke analysis as a response mechanism within an intrusion detection system. The combination of analysis techniques, placed within such an authentication/supervision framework has the potential to provide a significant improvement in system-wide security against impostor attacks, as well as ensuring transparency to legitimate end users.

6. REFERENCES

[1] Morrissey J.P.; Sanders P.W. & Stockel C.T. 1996. "Increased domain security through application of local security and monitoring"; Expert Systems; vol. 13; no. 4; pp296-305.

[2] Lunt T.F. 1990. "IDES: an intelligent system for detecting intruders"; Proceedings of the Symposium on Computer Security: Threat and Counter Measures"; Rome.

[3] Mukherjee B. & Heberlein L.T. 1994. "Network intrusion detection"; IEEE Networks; vol. 8; no. 3; pp26-45.

[4] Jobusch D.L. & Oldehoeft A.E. 1989. "A survey of password mechanisms: Weaknesses and potential improvements. Part 1"; Computers & Security; vol. 8; no. 7; pp587-603.

[5] Brown M. & Rogers S.J. 1993. "User identification via keystroke characteristics of typed names using neural networks"; International Journal of Man-Machine Studies; vol. 39; pp999-1014.

[6] Joyce R. & Gupta G. 1990. "Identity authentication based on keystroke latencies"; Communications of the ACM; vol. 33; no. 2; pp168-176.

[7] Legett J.; Williams G.; Usnick M. & Longnecker M. 1991. "Dynamic identity verification via keystroke characteristics"; International Journal of Man-machine Studies; vol. 35; pp859-870.

[8] Mahar D.; Napier R.; Wagner M.; Laverty W.; Henderson R.D. & Hiron M. 1995. "Optimizing digraph-latency based biometric typist verification systems: inter and intra typist differences in digraph latency distributions"; International Journal of Human-Computer Studies; vol. 43; pp579-592.

[9] Card S.K.; Moran T.P. & Newell A. 1980. "Computer text-editing: An information-processing analysis of a routine cognitive skill"; Cognitive Psychology; vol. 12; pp32-74.

[10] Umphress D. & Williams G. 1985. "Identity verification through keyboard characteristics"; International Journal of Man-Machine Studies; vol. 23; pp263-273.

[11] Legett J. & Williams G. 1988. "Verifying user identity via keystroke characteristics";International Journal of Man-Machine Studies; vol. 28; pp67-76.

[12] Bleha S.; Slivinsky C. & Hussein B. 1990. "Computer-access security systems using keystroke dynamics"; Actions on pattern analysis and machine intelligence; vol. 12; no. 12; pp1217-1222.

[13] Napier R.; Laverty W.; Mahar D.; Henderson R.; Hiron M. & Wagner M. 1995. "Keyboard user verification: towards an accurate, efficient, and ecologically valid algorithm"; International Journal of Human-Computer Studies; vol. 43; pp213-222.

[14] Furnell S.M.; Morrissey J.P.; Sanders P.W. & Stockel C.T. 1996. "Applications of keystroke analysis for improved login security and continous user authentication"; Proceedings of the 12th International Conference on Information Security (IFIP SEC '96), Island of Samos, Greece; 22-24 May, pp283-294.

[15] Ord T. & Furnell S.M. 2000. "User authentication for keypad-based devices using keystroke analysis"; Proceedings of the Second International Network Conference (INC 2000), Plymouth, UK, 3-6 July; pp263-272.

[16] Dowland P.S.; Singh H. & Furnell S.M. 2001. "A preliminary investigation of user authentication using continuous keystroke analysis"; Proceedings of the IFIP 8th Annual Working Conference on Information Security Management & Small Systems Security, Las Vegas; 27-28 September.

[17] Furnell S.M. & Dowland P.S. 2000. "A conceptual architecture for real-time intrusion monitoring"; Information Management & Computer Security; vol. 8; no. 2; pp65-74.

n experimental comparison of secret-based user uthentication technologies

I. Irakleous
Research Student, Network Research Group, Department of Communication and Electronic Engineering, University of Plymouth, Plymouth, UK
S.M. Furnell
Head of Group, Network Research Group, Department of Communication and Electronic Engineering, University of Plymouth, Plymouth, UK
P.S. Dowland
Lecturer, Network Research Group, Department of Communication and Electronic Engineering, University of Plymouth, Plymouth, UK
M. Papadaki
Research Student, Network Research Group, Department of Communication and Electronic Engineering, University of Plymouth, Plymouth, UK

ywords
nputer security, Effectiveness

stract
paper presents a comparative
dy of software-based user
hentication techniques,
trasting the use of traditional
sword and personal identifier
nbers (PIN) against alternative
thods involving question and
swer responses and graphical
resentation. All methods share
common basis of some secret
owledge and rely upon the
r's ability to recall it in order to
ieve authentication. An
erimental trial is described,
ng with the results based upon
participants. The alternative
thods are assessed in terms of
ctical effectiveness (in this
ntext relating to the
rticipant's ability to
thenticate themselves a
nificant time after initial use of
methods), as well as the
rceived levels of user
endliness and security that they
ovide. The investigation
ncludes that while passwords
d PIN approaches garner good
ings on the basis of their
isting familiarity to the
rticipants, other methods based
on image recall and cognitive
estions also achieved
fficiently positive results to
ggest them as viable
ernatives in certain contexts.

Introduction

One of the main aims of IT security is to ensure the availability of systems, while at the same time protecting them against unauthorised access, destruction and misuse. To this end, systems must control access to keep intruders and masqueraders out and permit access to the legitimate users. Although a variety of alternative techniques have been developed, using token-based authentication approaches (such as smart cards) or biometric solutions (such as fingerprint and facial recognition) (Sherman, 1992), the most common methods of authentication in current systems are based upon secret-knowledge approaches such as passwords and personal identification numbers (PINs). Despite their popularity, however, these methods are typically characterised as providing weak authentication, due mainly to the vulnerabilities introduced by end users (Jobusch and Oldehoeft, 1989). Common problems include the fact that many users forget their passwords, and compromise their protection by sharing them with other people. As such, it is appropriate to consider alternative methods of authentication that may overcome (or at least reduce) these problems, without introducing unnecessary complexity from the user perspective. Potential approaches here include altering the basis of the techniques away from purely recall-based approaches (which is the case with standard PINs and passwords), towards methods that rely upon less demanding concepts, such as recognition and provision of personal information.

A number of prior works have been conducted to enhance login security, whilst still retaining secret knowledge as the foundation of the approach. However, these approaches have typically been researched and evaluated on an individual basis, and it is desirable to evaluate them in a comparative study in order to obtain a more informed view of which, if any, are likely to represent acceptable alternatives. This paper presents the results of such an investigation, and begins by providing more details about the alternative methods. The discussion then proceeds to describe an experimental procedure by which five methods were evaluated in a practical trial. The results of this exercise are then presented, leading to consideration of the implications for practical systems.

Background

The predominance of password-based methods can largely be explained by the fact that they are conceptually simple for both systems designers and end users, and can provide effective protection if they are used correctly. However, users themselves often compromise the protection provided. Previous investigations have revealed a variety of problems, and typically include the fact that passwords are often: badly selected (and therefore more easily guessed or cracked), forgotten, written down, shared with colleagues, infrequently changed, and kept the same across multiple systems (Klein, 1990; Kessler, 1996). The work of Klein, for example, found that 25 per cent of the passwords (from a total sample of 15,000) were cracked after 12 months of exhaustive testing, with the help of a number of dictionaries including foreign words. More significant though is the fact that 21 per cent of the passwords (more than 3,000) were cracked in the first week, and 2.7 per cent of them were cracked in the first 15 minutes.

If the password approach is to be replaced or supplemented, then alternative means of

The research register for this journal is available at
http://www.emeraldinsight.com/researchregisters

The current issue and full text archive of this journal is available at
http://www.emeraldinsight.com/0968-5227.htm

MCB UP Limited
SSN 0968-5227]
DOI 10.1108/09685220210431854]

authentication are clearly required. Surveys have shown that fundamentally different approaches, such as using biometrics (authenticating users based upon their physiological or behavioural characteristics) or token based approaches (magnetic cards, smart cards) are not readily accepted by the user community, who for various reasons express a strong preference for the methods they already know (Furnell *et al.*, 2000). In addition, the financial cost associated with the introduction and maintenance of these other approaches will often preclude their adoption in many environments. For this reason, other approaches based upon secret knowledge, which do not incur any additional expenditure on hardware technologies, are considered desirable.

Previous research has highlighted the potential of question and answer based approaches, in which the user is asked to answer a series of questions, with correct answers leading to successful authentication. Clearly, such questions must require answers that are suitably distinctive to the legitimate user, in order to prevent everyone having similar answers or their responses being too easy to discover or guess. Such questions may be based upon cognitive or associative information (Haga and Zviran, 1991), as described later in the paper. The use of such questions has the potential advantage of using easily memorable (but nonetheless secret) information, but can involve a rather lengthy exchange between the user and the system in order to gain acceptance.

The solutions discussed so far have all been of a textual nature. However, given the transition to graphical user interfaces that has occurred during the last two decades, it is perhaps unsurprising that graphical authentication approaches have also arisen. For example, Blonder (1996) patented a graphical password in which the user can select a number of areas in a picture as a password. The weakness of this technique was that the user had to recall the location and the order of the regions. In another alternative, proposed by Jermyn *et al.* (1999), the "pasword" method was realised as a simple picture drawn on a grid. Other variations include the recognition of previously seen images, with an example being the D ej a Vu system (Dhamija and Perrig, 2000).

An experimental study of alternative methods

In order to enable a comparative study of alternative authentication methods, an experimental trial was devised incorporating five secret-knowledge based techniques. The methods selected were PINs and passwords (familiar methods, included to provide a baseline for reference), alongside two question and answer methods (using cognitive and associative questions respectively), and a graphical technique using an image-based PIN (hereafter termed imagePIN). The study sought to assess the practical effectiveness of the techniques, as well as friendliness and the perceived level of security from the user's perspective.

The effectiveness was gauged by means of a practical trial, using specially designed profiling and authentication systems to present the various techniques to a series of participants. Opinions relating to the friendliness and security of the methods were then obtained using a written questionnaire – completed by participants after they had participated in an authentication phase and witnessed their own performance using each technique. The construction of the experimental tools and the follow-up questionnaire are described in the subsections that follow.

The profiler

The profiler required each participant to identify him/herself and then provide appropriate responses for each of the methods under test. The profiling procedure for each of the methods is summarised below:

1 *Passwords and PINs.* The implementation of these methods was fairly standard, with each participant being asked to supply a four digit PIN and a password of at least eight characters. Participants were requested not to select a password or PIN that they already used on other systems, as the aim of the exercise was to assess their ability to recall new details, and thereby put these more familiar methods on an equal footing with the other techniques when it came to assessing ease of information recall. Nonetheless, as later results will indicate, some participants did not follow this guideline.

2 *Cognitive questions.* Participants were asked to provide answers to a series of 20 questions, each requiring factual or opinion-based answers. The questions requested information that was personal to the participant, and would therefore be difficult for a potential masquerader to guess in an operational scenario. The questions used are listed:
- What is your mother's maiden name?
- Where were you born?

- What is your favourite colour?
- What was the name of your best friend at school?
- What is your favourite music?
- What is your favourite food?
- What was the name of your first pet?
- Which primary school did you go to?
- What is your favourite sport?
- Where was your first house?
- What make was your family's first car?
- How old were you when you had your first kiss?
- What is your favourite film?
- Where was the first place you remember going on holiday?
- What was your favourite subject at school?
- What is the most important part of your body?
- What is your favourite type of animal?
- What is the name of your favourite relation?
- How many cousins do you have?
- What is your favourite shape?

Even in cases where the participants might not have had a genuine answer (e.g. they may never have had a pet), it was expected that they would still be able to provide a response that could later be reproduced if prompted to answer that question.

3. *Associative questions*. Participants were then asked to provide word association based responses to a set of 20 keywords. The keywords:
   - blue;
   - house;
   - table;
   - computer;
   - friend;
   - peace;
   - glass;
   - marriage;
   - sea;
   - love;
   - cat;
   - music;
   - fire;
   - seven;
   - video;
   - father;
   - food;
   - remote;
   - fast; and
   - door.

   They were carefully chosen to ensure that a number of different responses were theoretically possible in each case.
4. *ImagePIN*. The user had to select five images from a number of icons, by clicking on them with the mouse. Later authentication would work by the user reselecting the same images in the correct sequence.

The user interface of the profiling system is illustrated in Figure 1.

After the profile had been created, a short training exercise was performed using the second program, the authenticator, in order to familiarize the users with how the later authentication test would work. After this, it was up to the participants to attempt to remember the details they had provided in order to perform the later authentication tests.

The authenticator

The authentication tests took place one month after the initial profiling, with the aim of assessing whether the participants were able to adequately recall the information that they had previously provided during profiling and thereby authenticate themselves successfully. The interface of this system was very similar to that of the profiler, and two aspects are illustrated in Figure 2.

In the case of the PIN, password and imagePIN methods, the participant was directly asked to provide the same information as originally profiled. For the cognitive and associative methods, however, they were asked to answer five randomly selected questions out of the 20 that had been profiled in each case. This was considered to represent a good simulation of how such question and answer authentication techniques would be implemented in practice.

Participant questionnaire

Following the authentication test, all the participants were asked to complete a questionnaire, in order to determine their regular exposure to user authentication methods in other contexts and to assess their views about the different methods under trial. The following key elements of information were collected:

- the number of different passwords they use;
- the frequency with which they use PINs and passwords;
- whether or not they use the same password in their applications;
- the composition of their password(s) (e.g. letters, numbers, symbols); and
- ranking the trialed methods according to the perceived user friendliness, level of security, and overall preference.

A total of 27 participants were involved in the profiling and subsequent authentication testing, and the results of the study are described in the next section.

Irakleous, S.M. Furnell,
.S. Dowland and M. Papadaki
n experimental comparison
f secret-based user
uthentication technologies
formation Management &
omputer Security
0/3 [2002] 100–108

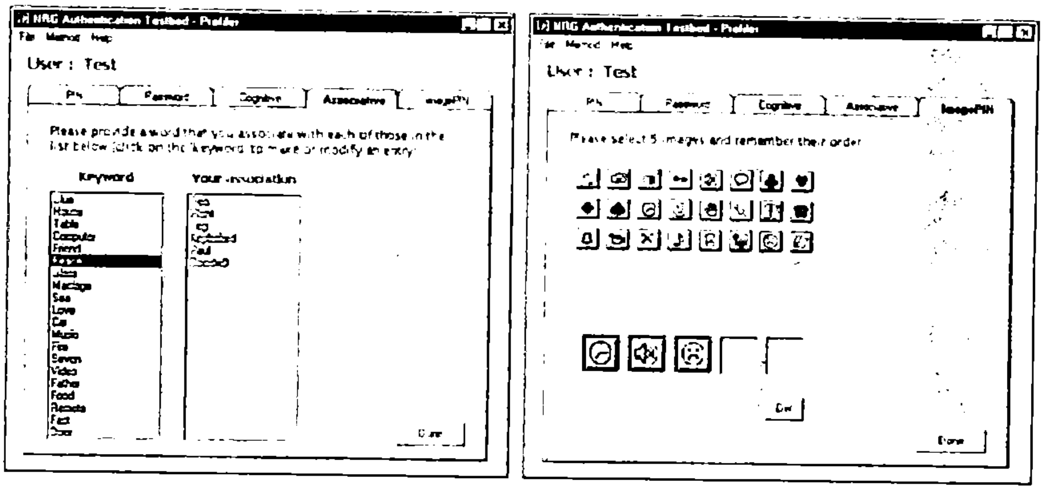


Figure 1
Profiler system (showing associative questions and imagePIN screens)

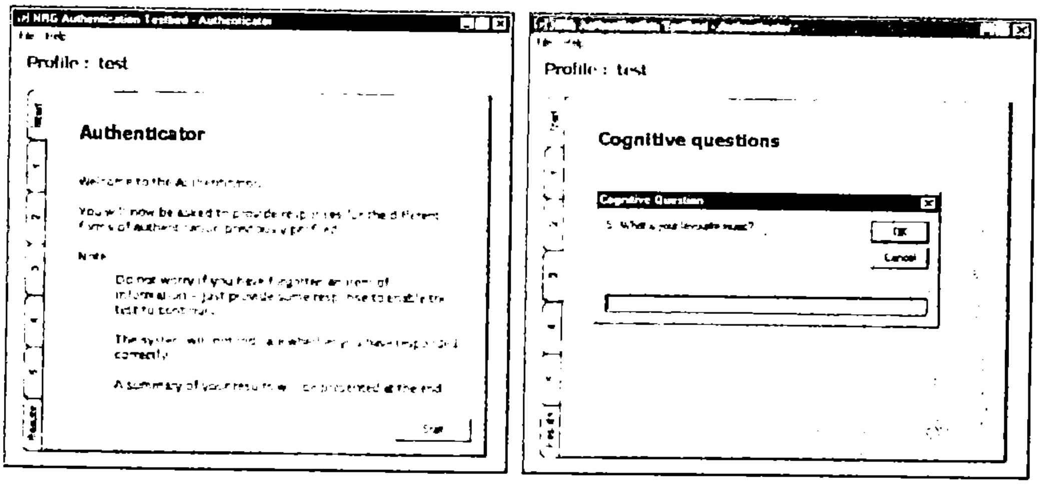


Figure 2
Authenticator system (showing welcome and cognitive question screens)

Experimental results

The results presented here encompass the effectiveness of the techniques (in terms of user recall) that was observed in the practical trial, as well as the participants' subsequent opinions in relation to the methods. It should be noted that, in the discussions and graphs that follow, the percentage figures have been rounded to whole numbers.

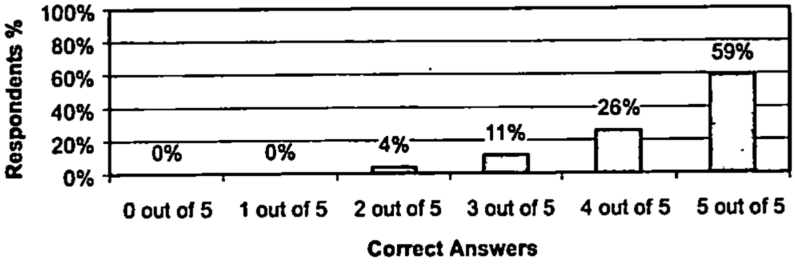
In order to gauge their current exposure to authentication techniques, the participants were asked how many different passwords they have to remember and how often they use them. Only 4 per cent of the participants had just a single password or PIN to remember, whilst 59 per cent claimed to have up to five, 22 per cent claimed to have between five and ten, and 15 per cent claimed to have in excess of this number. In terms of their frequency of use, 85 per cent claimed daily usage, while 11 per cent indicated once every two days, and 4 per cent claimed three to five times a week. No one indicated that they used PINs or passwords on a less frequent basis than this. It can be concluded from these findings that, although the overall sample of users was small, the participants all had considerable experience of using traditional authentication methods and were therefore suitably qualified to participate and comment on this study.

The practical evaluation began by examining the participants' performance in relation to the password and PIN methods. The results indicated that 70 per cent of the participants had succeeded in authenticating themselves using passwords, and a similar proportion (67 per cent) were successful using the PIN based method. Although these results initially appear very encouraging from the perspective of the participants being able to accurately recall the details after an absence of a month, the results of the accompanying survey revealed that a significant number of people had not followed the request to use different passwords and PINs than the ones normally used in other applications. In fact, only 56 per cent used different passwords and 41 per cent used different PINs. Within these subgroups, the authentication success was markedly lower – 53 per cent of them succeeded in the password test and only 36 per cent in the PIN version. By contrast, within the subgroups that used the same details as in other systems, 92 per cent of them succeeded with passwords and 87 per cent succeeded with PINs, so these figures can be considered to have artificially inflated the overall results.

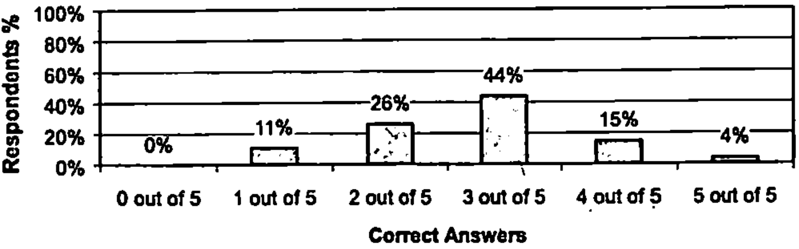
In the cognitive and associative question tests, the participants were presented with a random selection of five questions out of the 20 that they were profiled for. Authentication was judged to be successful if all five questions were answered correctly. With the cognitive questions, a success rate of 59 per cent was observed, whilst a number of further participants did succeed in answering a proportion of the questions presented to them. The distribution of correct answers in the cognitive test is shown in Figure 3.

With the associative questions, the success rate was significantly lower. Only 4 per cent (equivalent to one person) managed to correctly answer all five questions and the distribution of correct answers across five random questions is shown in Figure 4.

igure 3
istribution of correct answers in cognitive questions



igure 4
istribution of correct answers in associative questions



These results suggest that the associative question method is extremely problematic in relation to the correct recall of the information, and that participants are inconsistent in the words that they most readily associate with the keyword prompts. A further problem observed in the results of this study was that many participants chose the same associations for certain keywords, suggesting that the method could be easily targeted for masquerade attacks if used in practice. Table I summarises the cases in which the same associations were chosen for each keyword. The highest frequency of duplication was 44 per cent, in which respondents had chosen the word "control" as the associative response to the keyword "remote".

For the final technique, the imagePIN, the participants had to recall their graphical PIN by reselecting the original icons in the correct order, with 63 per cent being successfully authenticated. Even though the implementation of the method offered the participants the opportunity to undermine the security by selecting the same icon five times, only two participants actually did this.

Figure 5 summarizes the overall results of the authentication tests, indicating the percentage of respondents who would have been successfully authenticated using each of the methods.

Having experienced the techniques and witnessed their own performance, the participants were asked to rate the approaches on the basis of user-friendliness, security, and overall preference.

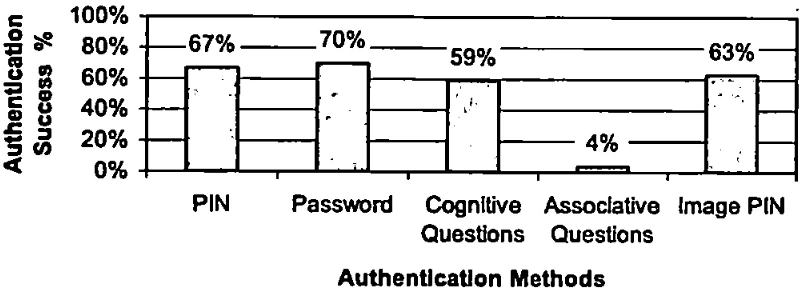
In terms of user-friendliness, participants were asked to assess the methods on a five-point scale, progressing from "easy" to "hard". The best outright indicator of

Table I
High frequency associative responses

Keyword	Frequent word associations
Blue	Sky (41%), Sea (15%)
House	Big (15%)
Table	Food (22%)
Computer	Work (11%), Game (7%), Internet (7%)
Peace	War (15%)
Glass	Wine (22%), Broken (11%)
Sea	Blue (11%)
Love	Hate (11%), Marriage (7%)
Music	Rock (15%), Dance (7%)
Fire	Red (11%), Alarm (11%), Engine (7%)
Seven	Film (15%), Seven (7%), Days (7%)
Video	Games (11%), Movie (11%), Tape (7%)
Father	Mother (19%), Names (15%)
Remote	Control (44%)
Fast	Food (22%), Car (19%)
Door	Key (11%), Open (11%), Closed (7%)

preference in this case was where methods were ranked as "easy". In this context, passwords were ranked first, receiving 48 per cent, followed by the PIN method with 44 per cent. The third position was shared by the cognitive question and imagePIN methods, with 22 per cent respectively. Last was the associative method with only 4 per cent. Taking a wider view, and considering the total percentages for which methods were rated "medium" or above, the password was still favourite, with 96 per cent, followed by the PIN with 93 per cent, cognitive questions with 81 per cent, the imagePIN with 59 per cent, and associative questions with 48 per cent. Looking from this viewpoint serves to place some separation between the cognitive and imagePIN methods, and shows that more people tended to express concern over the friendliness of the latter technique. The full results are presented in Figure 6.

igure 5
uthentication methods success



The second ranking addressed the perceived level of security. In this case, the password still fared well, with a combined total of 78 per cent rating it to offer a "medium" to "high" level of protection. In this instance, however, the popularity was also equalled by the cognitive and imagePIN methods (and it can be noted that both of these methods actually exceed the results for passwords if only the "high" and "medium high" ratings are considered). Meanwhile, the PIN method attained 53 per cent, and the associative approach was again ranked lowest, with 45 per cent ranking it in the "medium" to "high" range. Figure 7 presents the perceived level of security for each authentication method.

The final question asked the participants to rank the methods according to their overall preference. Password method was again the most preferred form of authentication, with 44 per cent, as shown in Figure 8. In the second place is PIN method with 22 per cent, and third is the imagePIN method with 19 per cent. It is therefore clear that the more traditional and familiar methods of authentication are still the most readily accepted. However, if the rationale behind the alternative methods is accepted (i.e. that passwords and PINs are open to compromise), then it is relevant to give further consideration to the results and responses in the other categories.

Figure 6
Perceived user-friendliness

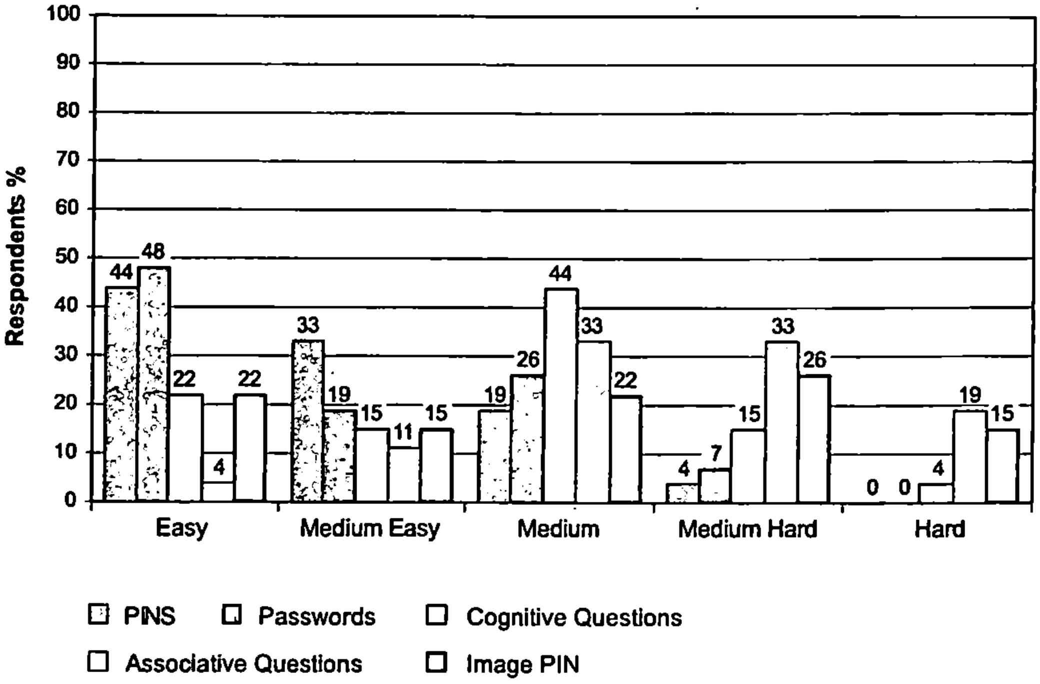


Figure 7
Perceived security

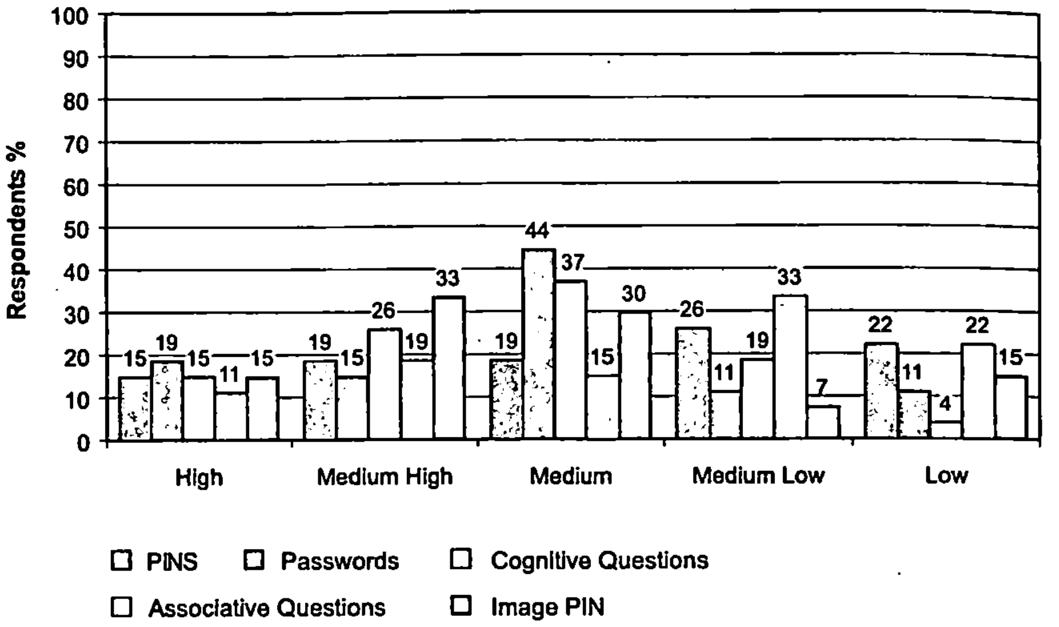
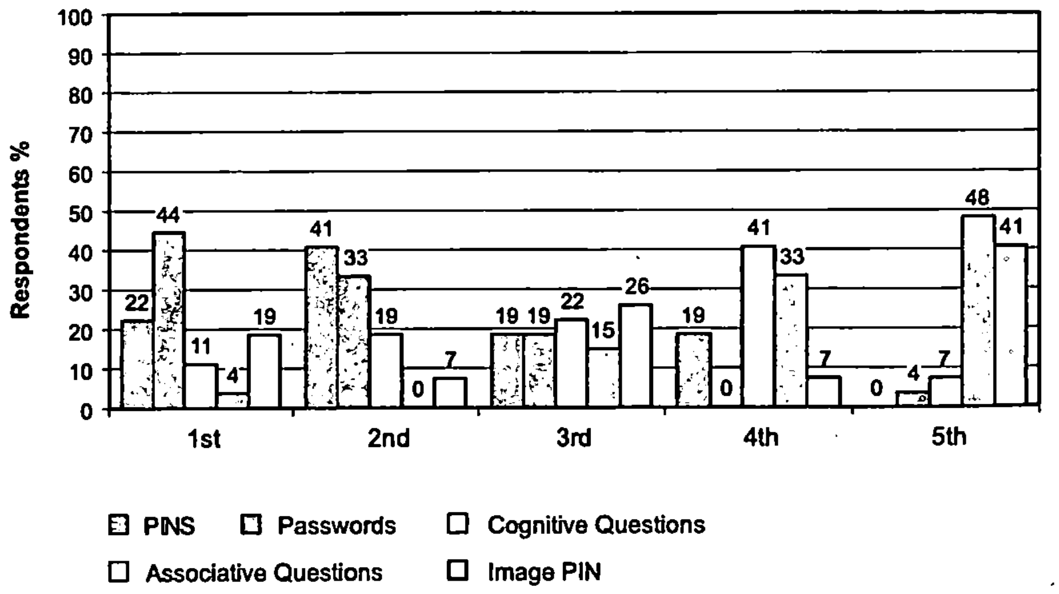


Figure 8
Overall preference of trialed methods



Discussion

Although people clearly prefer passwords and PINs, the other results obtained continue to suggest concerns about the level of security they actually provide. For example, analysis showed that 48 per cent of the participants selected passwords that might be easily guessed or cracked (e.g. based upon dictionary words, variations of their name, or foreign words written in English characters). Only 38 per cent of participants used an alphanumeric combination, and fewer still (4 per cent) introduced other symbols into their passwords. These results increase the attractiveness of the other methods, which may be less vulnerable to such unintentional compromise.

The participants' performance in relation to the cognitive questions was relatively strong, with 59 per cent successful authentication (interestingly, a previous study by Haga and Zviran (1991) reported better results, with 74 per cent, for a broadly similar set of cognitive questions). A further point noted about the cognitive technique was the relatively time consuming nature of the profiling phase, in which the participants had to provide answers for all 20 questions. In addition, several participants expressed concern about the nature of the information that was requested, and were reluctant to provide genuine answers to the questions during the trial for fear that the information might be accidentally divulged. Particularly notable questions in this respect were in relation to mother's maiden name (a commonly used identity verification question in other contexts, such as bank accounts), place of birth, and age of first kiss. Overall, however, this method was ranked relatively high in terms of perceived user-friendliness and security.

The associative approach proved to be weak as an authentication method, with the performance of the participants (only 4 per cent success) suggesting that it cannot deliver an adequate level of effectiveness. It is considered that this poor performance can in part be explained by the fact that users still have to remember potentially abstract information (as opposed to the more recognition-oriented approaches of cognitive questions), placing more or less the same demand on their memory as the password method. In addition, the results raise questions over the level of security that the approach would provide – the fact that many participants chose the same word associations suggests that the method would be vulnerable to attackers attempting to guess the likely associations. At the very least, this requires that more care must be taken in the selection of the keywords, to ensure that none of them have obvious first-choice answers. It may again by observed that a previous study of the same basic method reported a far higher success rate, with an overall average 69 per cent recall after a period of three months (Haga and Zviran, 1991). It must be noted, however, that there was a significant difference in the experimental procedure in this case, as participants were asked to select their own keywords, as well as the appropriate associative responses.

The imagePIN approach demonstrated positive results in the authentication phase, with 63 per cent success, placing it very close to the results observed for passwords. This result is partially explained by the findings from previous surveys, which have shown that people tend to have less difficulty in recognising previously seen pictures than they do in recalling passwords or phrases from the memory (Bensinger, 1998; Sasse *et al.*, 2001). In addition to its practical effectiveness, the imagePIN scored well in terms of user acceptance, which bodes well for the rating that it might receive if users were given additional time to familiarise themselves with it. Another point worth noting is that the imagePIN method as implemented for the study was rather crude, with a set of standard Windows icons having been used as the selection of available images. With more consideration given to the number and range of images available, it is likely that the perceived user-friendliness of the approach could be further improved. Having said this, there was also a fairly high proportion of respondents who put it as their clear least favourite, whereas most of the other methods did not elicit such strong negative opinions.

Although some techniques suggested themselves as potential alternatives to standard passwords and PINs, it does not necessarily follow that they would make good replacement methods in all contexts. For example, the use of cognitive questions could potentially be too time consuming as a regular means of login authentication. The technique could, however, provide a good secondary level of authentication, which could be invoked in a number of scenarios (e.g. when a user tries to perform a sensitive activity, in response to a suspected masquerade attack, or simply at random intervals). Image based authentication techniques could be more easily implemented as an initial login technique, but their applicability would be limited to systems that are able to offer sufficient graphical displays. This would, currently, rule out devices such as mobile phones (where standard PIN methods currently predominate), but could still usefully include other PIN-based devices such as personal digital assistants and automated teller machines.

Conclusions

The paper has presented a comparative study of five user authentication techniques based upon secret knowledge. With the clear exception of the associative approach, the practical effectiveness of the techniques was closely comparable. However, in terms of the overall preference, the known and familiar methods of passwords and PINs were, perhaps unsurprisingly, favoured. Having said this, if the previous arguments and evidence regarding the weaknesses of these methods are accepted, then it may be reassuring to consider that the cognitive and imagePIN methods are already comparably effective from a user recall perspective, and given further training and exposure these methods may gain greater acceptance.

Although the initial results are encouraging, two significant aspects were not addressed by the work to date. Firstly, the judgements relating to user-friendliness of the methods were based on a relatively brief level of exposure in the case of the question and answer approaches and the imagePIN method. A longer-term trial is therefore required in which participants use the alternative methods in day-to-day operations, in place of their normal passwords or PINs. This will allow a more accurate impression to be gained regarding the perceived user-friendliness. The second aspect that requires attention is the level of protection that the new methods actually deliver when compared to the traditional approaches. The study described here did not attempt to assess the ability of participants to successfully masquerade as other users – although the duplication of responses that was observed for the associative questions would suggest that this would clearly be possible. As such, the methods need to be assessed in terms of their susceptibility to compromise by informed parties (e.g. those who know the person they are trying to impersonate, and may therefore be able to determine the correct cognitive and associative responses) and by simple guesswork. A further aspect that is worthy of investigation (in relation to the imagePIN, or indeed other graphical approaches) is how well users are able to cope with multiple sets of login information. As earlier results confirmed, users today have to remember multiple PINs and passwords, and it is therefore relevant to know whether remembering multiple imagePINs serves to simplify the issue or complicate it further. These aspects represent the focus of ongoing investigation by the authors.

References

Bensinger, D. (1998), "Human memory and the graphical password", available at http://www.passlogix.com/.

Blonder, G. (1996), United States Patent 5559961.

Dhamija, R. and Perrig, A. (2000), "D ej a Vu: a user study using images for authentication", SIMS/CS, University of California, Berkeley, CA.

Furnell, S.M., Dowland, P.S., Illingworth, H.M. and Reynolds, P.L. (2000), "Authentication and supervision: a survey of user attitudes", *Computers & Security*, Vol. 19 No. 6, pp. 529-39.

Haga, W.J. and Zviran, M. (1991), "Question-and-answer passwords: an empirical evaluation", *Information Systems*, Vol. 16 No. 3, pp. 335-43.

Jermyn, I., Mayer, A., Monrose, F., Reiter, M.K., and Rubin, A.D. (1999), "The design and analysis of graphical passwords", *Proceedings of the 8th USENIX Security Symposium*, August 1999.

Jobusch, D.L. and Oldehoeft, A.E. (1989), "A survey of password mechanisms: part 1", *Computers & Security*, Vol. 8 No. 7, pp. 587-604.

Kessler, G.C. (1996), "Passwords – strengths and weaknesses", January 1996, available at http://www.garykessler.net/library/password.html

Klein, D. (1990), "Foiling the cracker: a survey of, and improvements to, password security", *Proceedings of the 2nd USENIX Security Workshop*, August 1990.

Sasse, M.A., Brostoff, S. and Weirich, D. (2001), "Transforming the 'weakest link' – a human/computer interaction approach to usable and effective security", *BT Technology Journal*, Vol. 19 No 3, pp. 122-31.

Sherman, R. (1992), "Biometrics futures", *Computers & Security*, Vol. 11 No. 2, pp. 128-33.

Enhancing response in intrusion detection systems

M.Papadaki, S.M.Furnell, S.J.Lee, B.M.Lines and P.L.Reynolds

Network Research Group,
Department of Communication & Electronic Engineering,
University of Plymouth, Plymouth, United Kingdom
Email: info@network-research-group.org

Abstract:
With rising levels of attacks and misuse, intrusion detection systems are an increasingly important security technology for IT environments. However, while intrusion detection has been the focus of significant research, the issue of response has received relatively little attention. The majority of systems focus response efforts towards passive methods, which serve to notify and warn, but cannot prevent or contain an intrusion. Where more active responses are available, they typically rely upon manual initiation. The paper examines the reasons for this, and argues that a more comprehensive and reliable response framework is required in order to facilitate further automation of active responses. A range of factors are identified that a software-based responder agent could assess in order to improve response selection, and thereby increase trust in automated solutions.

Introduction

An increasing level of attacks upon IT systems represents a seemingly unavoidable reality of the Internet revolution. From the malicious activities of external hackers to deliberate misuse by organisational insiders, no sector has shown itself to be immune from attack, and the provision of a public-facing server is effectively all that an organisation needs to do in order to establish itself as a potential target. Evidence of the problem is provided by results from the annual CSI/FBI Computer Crime and Security Survey, which has seen the percentage of respondents reporting incidents rise from 42%, in 1996, to 64% in 2001 (Power 2001), having reached an interim highpoint of 70% in 2000. The associated financial losses have also increased, and the 2001 survey results reported total losses approaching $378 million (from 186 respondents who were willing and able to quantify the financial impacts of their incidents). In fact, since 1997, the annual damage from security breaches has been increasing by an average of 52% every year.

Statistics such as those above emphasize the need for security in networked systems; and a key technique for combating attacks is provided by Intrusion Detection Systems (IDS). The concept of intrusion detection was originally proposed by Denning (1987), with the underlying rationale that the complicated infrastructures of computer and network systems are inherently insecure, and thus may be under attack. Pursuing the aim of creating totally secure systems may not be feasible or cost effective, so an intrusion detection system should be able to detect such attacks, preferably in real time. Since Denning's original work, the concept has received wide acceptance in the computer security domain, and several efforts have concentrated on the development of practical intrusion detection systems.

ISSN 1445-3312 print/ISSN 1445-3347 online

There are many challenges in that process and, to date, the focus of research has been on the detection capability of systems (Mukherjee et al. 1994;2000). However, the issue of response to detected incidents is another significant issue, but has so far been largely overlooked (Schneier 2000) and therefore requires further research in its own right.

The paper begins by introducing the concept of intrusion response, and considering the different approaches by which it may be realised. The response capabilities of current intrusion detection systems are then analysed, from the perspective of both commercial products and ongoing research projects. The need for further enhancement is identified, leading to the proposal of a broader response framework, and the identification of various contextual factors that need to be considered in order to select appropriate responses.

The concept of intrusion response

Intrusion response can be defined as the process of counteracting the effects of an intrusion. In the context of intrusion detection it includes the series of actions taken by an IDS following the detection of a security-related event. It is important to note that consideration is not only given to taking action after a full-scale breach has been detected, but also when events of interest take place and raise the alert level of the system (i.e. the early stages of an attack, when the system is suspecting the occurrence of an intrusion, but is not yet sufficiently confident to take action).

In general, the aims of response actions can be classified into one of the following categories:

1. Notification about the occurrence of an intrusion.
2. Protection of system resources:
   - in the short term, this will include mechanisms to contain the intrusion, as well as to recover and restore the system to a well known state.
   - in the longer term, it includes learning from the intrusion, using this knowledge to remove identified vulnerabilities of the system, and to enhance the detection and response capability. The objective here is to prevent reoccurrence of the intrusion.
3. Identification of the perpetrator of the intrusion.

At the highest level, there are two main approaches to intrusion response; namely human/organisational approaches and technical methods. The former are those that involve human processes and organisational structures, and may include actions such as reporting an incident to the police or invoking disciplinary procedures (e.g. in cases where internal personnel are responsible). From the list above, the process of identifying the perpetrator often requires further investigation and co-operation with other parties, such as Incident Response Teams, and thus it naturally falls under the human/organisational aspect of response. By contrast, technical responses involve the use of functional techniques and software-based methods. These technical actions can themselves be further sub-classified, into either passive or active forms of response (Bace and Mell 2001). Technical response actions can also be characterised as either manual or automated, according to the way they are initiated (Amoroso 1999). The main distinctions will now be considered in more detail.

Passive and Active Responses

Passive responses aim to notify other parties about the occurrence of an incident, relying on them to take further action.

Passive responses may include methods such as:

- Recording details for later inspection (e.g. adding an entry in a log file);
- Alerting an administrator, by displaying a pop-up window on the console, or generating an email, pager or mobile phone message;
- Generating alarms and alerts to report to a central network management console by using SNMP traps and messages.

Passive responses, in the form of notifications and alerts, have been used by IDSs since their initial development, primarily as an indicator of their detection effectiveness. Hence they are still present in every intrusion detection product, offering the standard level of response, and making them the most common response option in commercial IDS systems. The obvious disadvantage here is that they do nothing to impede the intruder, and rely upon someone to manually respond at some later point (by which time it may be too late to avert a more significant security breach).

In contrast to the passive approaches, active responses are the actions taken to counter the incident that has occurred. Such actions might include the following approaches:

- collecting more information about the incident (e.g. issuing an authentication challenge, increasing the monitoring level);
- limiting permitted user behaviour or process activity;
- blocking network traffic through firewalls and routers;
- terminating network connections;
- introducing delay on network connections.

Active responses can have a more significant impact upon a system, and thus they engender the danger of causing unwanted effects, in the event that they are falsely initiated. In order to overcome this danger, careful consideration should firstly be given to the thoroughness and extensiveness of the response options available. It is also important to study the conditions under which the selection of appropriate responses is made. This requires consideration of the factors that can influence the response decision process and assessment of their weighting upon that process.

Not surprisingly, active responses have mainly been used in research prototype systems. Although there are some commercial systems utilising active response methods, especially ones that involve blocking of network traffic and termination of network connections, their application is still in an early stage and their effectiveness has not yet been conclusively proven.

Manual and Automated Responses

The detection of a suspected intrusion typically triggers a manual intervention by a system administrator, after having received an alert message from the intrusion detection system. The IDS can additionally assist the incident response process, by providing the details of the attack, saved in a log file (Bace 2001). However, responding manually to intrusions is not necessarily an easy task, as it can represent a significant administrative overhead. That may involve dealing with a high number of alerts and notifications from the IDS, ensuring awareness of security bulletins and advisories from incident response teams, and taking appropriate actions to resolve each of the alerts reported. From the system administrator's perspective, the main requirement is to ensure that the system remains operational and available – this is what the users expect and complaints will quickly occur if this is not the case. Unless resolving a reported incident is explicitly required to ensure that this is the case, then the task is likely to be given a lower priority.

The ability to mount a rapid response to an attack is, however, extremely important. The effect of reaction time on the success rate of attacks was demonstrated by Cohen, who carried out a simulation of attacks, defences and their consequences in complex cyber systems (Cohen 1999). The results indicated that if skilled attackers are given 10 hours after they are detected, and before a response is generated, then they will be successful 80% of the time. If they are given 20 hours, they will succeed 95% of the time. At 30 hours, the attacker almost never fails. The results also indicate that if a skilled attacker is given more than 30 hours, the skill of the system administrator will make no difference, as the attacker will irrespective of that succeed. On the other hand, if the response is instant, the probability of a successful attack against a skilled system administrator is almost zero. This proves that there is a relationship between the effectiveness of response and the time it is issued, and that there is a window of opportunity for an attacker if response is not issued on time.

Another factor that highlights the need for automated response is the changing nature of the techniques employed by attackers, including the widespread use of automated scripts to generate attacks of distributed nature (Cheung and Levitt 1997). These can further diminish the ability to respond manually, since there is practically no time available to do so.

At the time of writing, the degree of automation in current intrusion detection systems is very low, being largely limited to the automation of passive responses. Nonetheless, a feasibility-level research study has estimated that 33% of available response actions have the potential to be safely automated, without having to further enhance the detection capability of the IDS (Lee 2001). As such, this would have the potential to significantly reduce the burden upon system administrators.

Response capabilities of current IDS

A literature search was carried out to investigate the response capability of current IDSs, focusing upon the systems with more interesting approaches to intrusion response. The aim of this task was to cover the most representative set of response options available, rather than reflect the degree to which automated active response has been adopted in the intrusion detection domain. Thus, consideration is given to the more significant response features of commercial IDS products available (Table 1) followed by the systems under research.

As expected, nearly all Intrusion Detection Systems offer a wide range of passive responses, but the situation regarding active responses is somewhat more varied. There are systems that do not offer any active methods, while others that seem to offer a wide range of options. However, it seems that the active responses (terminate/reset network connections, block network traffic) available for network-based IDSs are more widely adopted than the ones fitted for host-based systems (limit permitted user behaviour), suggesting an opportunity for further enhancement.

		Passive Response			Active Response				
IDS name	NB / HB*	Pop-up window alerting	E-mail, Pager alerting	SNMP (trap) alerting	Collect more information	Limit permitted user behaviour	Terminate / reset network connection	Block network traffic	Introduce Delay
Axent Technologies NetProwler (Hurwitz Group 2000)	NB	✓					✓	✓	
Axent Technologies Intruder Alert (Shipley 1999)	HB	✓	✓	✓					
CentraxICE ICEpac, BlackICE (Gilliom 2001)	HB / NB	✓	✓	✓		✓	✓		
Cisco Secure (Cisco Syst. 1998)	NB	✓	✓		✓		✓	✓	
eTrust IDS (Computer Associates 1999; 2001)	NB	✓	✓	✓	✓			✓**	
ISS RealSecure 6.0 (ISS 2001)	HB / NB	✓	✓	✓		✓	✓	✓	

* NB / HB: Network-based / Host-based

** Blocking of network traffic is not done via reconfiguration of a router or firewall, but by using patent pending "***unobtrusive blocking***" based on pre-defined rules (*stateful dynamic blocking*) or in response to a specific alert.

Table 1: Response capabilities of current IDSs

In addition to commercial products, there is also a significant amount of active research in the IDS domain. As such, it is relevant to consider whether these have more advanced approaches in the context of response.

Emerald

Event Monitoring Enabling Responses to Anomalous Live Disturbances (EMERALD) is an intrusion detection project being pursued within the Systems Design Laboratory at SRI International. Within the Emerald architecture, the Resolver is a countermeasure decision engine capable of handling the alerts from its associated analysis engines and invoking response handlers to counter malicious activity. The Resolver is an expert system that receives the intrusion and suspicion reports from the profiler and signature engines. Based on the combination of these with reports from other peer monitors, it decides which response to invoke, and how to invoke it. Possible responses may include direct countermeasures such as closing connections, terminating processes, or the dispatching of integrity-checking handlers to verify the operating state of the analysis target (Porras and Neumann 1997). EMERALD provides an interesting architectural approach, focusing on the co-operation of distributed response elements. However, although elements of the detection functionality have been realised in practice, the eResponder (the countermeasure invocation system) is still under development at the time of writing (SRI 2001), and the conceptual definition that has been published does not specify in detail the response mechanism, the actions available, or the operation of the decision engine.

Response and Detection Project

This research project is being pursued as a collaboration between Boeing Corporation, Trusted Information Systems, and UC Davis University (UC Davis 2000). It is an effort to combine state of the art intrusion detection systems with intelligent firewalls and routers to form an Intranet wide automated response system. The basic idea is to enable cooperation among response components in a virtual security network, where security components alert each other of an attack, and a component will be selected to initiate an automated response. Automated responses mainly examine network-based attacks, and at present are restrained to filtering network packets. However, focus has been given to the extension of response options, including options such as the introduction of delay into network connections and the replacement of sensitive files with look-alikes.

Adaptive, Agent-based, Intrusion Response System (AAIRS)

The AAIRS project is being pursued within the Computer Science Department of Texas A&M University (Carver et al. 2001). It focuses on the response decision mechanism, proposing a methodology for adaptive automated response using intelligent agents. In AAIRS, a new *Analysis agent* is created every time a new attack is reported by the IDS. In collaboration with other agents (for response and policy) an abstract response plan is developed, taking account of any appropriate legal, ethical, institutional, or resource constraints specified in the policy. The plan is then passed to a *Tactics agent*, which decomposes it into specific actions and invokes the appropriate components of a *Response Toolkit*. Both the *Analysis and Tactics agents* employ adaptive decision-making based on the success of previous responses, which serves to limit uncertainty in the response decision process, and facilitate adaptation of the system based on the effectiveness of its detection and response capability in the past (Ragsdale et al. 2001). For the response decision process, the following factors have additionally been taken into account (Carver and Pooch 2000):

- Timing of the attack (pre-emptive, during attack, after attack)
- Type of attack (threat to confidentiality, integrity, availability)
- Type of attacker (Cyber-gangs, Economic Rivals, Military Organisations, ...)
- Degree of suspicion (high, ..., low)
- Attack implications (Low implications, ..., Critical implications)
- Environmental Constraints (No offensive responses, ..., No router Resets)

Of the response-oriented work described in published literature, the AAIRS project is considered to offer the most comprehensive treatment of the issue to date, giving considerable focus to the mechanism, and the influencing factors, of the response decision process.

Limitations of current intrusion response methods

An important issue that was not reflected in Table 1 was the degree to which the response methods are automated. Although it is technically feasible to implement and automate many forms of response in software, it is not as straightforward a solution as it might seem. Whereas passive responses have already been automated to a large extent, active responses are largely initiated manually. The reason for this is that passive actions have little impact upon a system, and thus there is no danger of causing damage if a response is initiated in a false positive alarm scenario. On the contrary active responses could cause disruption to legitimate users, affect their access level to the system or even cause an unintentional denial of service attack to the system itself. Hence we need to make sure that when response actions are launched automatically - without prior human authorisation - they have the intended effects and do not put the system in a greater risk than it currently is. This requires confidence in the detection capability of a system (i.e. that its assessment of a scenario as being intrusive is accurate), as well as its subsequent ability to choose appropriate countermeasures in response.

Even if automated response capabilities were to be made widely available, there is a question of whether they would actually have practical value in the eyes of security administrators, who may prefer to place trust in their own abilities rather than those of the system. This viewpoint is reflected in the results of an email survey conducted by Lee (2001), in which various IDS vendors and intrusion detection specialists were asked to comment upon the automated response issue. A number of relevant responses are presented below (having been anonymised to remove details of specific individuals or products). It can be seen that whereas some vendors do not support automated response, but at the same time do not exclude the option of doing so, other commentators can see a fundamental risk in the concept:

> *'We do not have any self-imposed automated proactive responses; we enable the creation of policies for response.'*
>
> *'We have the functionality for automated responses but we haven't yet explored fully what we should do to pro-actively deal with suspected intrusions.'*
>
> *'Our product at this time does not provide a particular response mechanism out of the box,... What I have found is that, at least in North America, many of the security professional prefer not to use automated response systems.'*
>
> *'We think it is dangerous to put all your faith in automatic responses believing that you are protected. Attackers are very smart and know how to use your own equipment against you if it will benefit their attack. ...We believe that the ability to see if an attack was successful or not and then have a human acting on that is better for the overall health and security of the network. ...most people in the security product industry agree that automatic responses can be very dangerous and should not be relied upon to make important decisions about your networks.'*
>
> *'Proactive measures are a reasonable idea, unless they can be subverted. For instance, if you decide to shut down your network connections as a proactive approach, then an intrusion attempt can be used as a denial of service...'*
>
> *'Retaliation (DoS for instance) is out of the question since one can't ignore the impact on innocent users coming from the same network (say with an ISP). Therefore, [DELETED] will continue to only support detection and reporting.'*
>
> 'Right now, in its current form, I don't believe that the current products are mature enough to be performing active response. ... any device that is re-configuring infrastructure equipment (shunning) could easily be turned into a denial of service tool.'

Although the negative comments above can be considered to offer a valid perspective, what they tend to overlook is the previous argument that, in many circumstances, manual response may not represent a viable alternative (e.g. in the context of attack via automated scripts). As such, it can be concluded that, in spite of the difficulties, efforts are required to improve the prospects of automated methods.

Extending Automated Response

In order to address the automated response issue, it is considered that further attention is required in two main areas; namely the broadening of possible response options, and the assessment of factors that influence the suitable response to be taken. Appropriate attention to these aspects will help to address the problem of reliability in relation to automated responses.

Broadening of response options

Further consideration should be given to novel response actions, which will possibly have less significant impact upon the system and its legitimate users, causing at the same time the desired effect upon the attack, and preserving the uncompromised state of the system. Intuitively, however, these requirements may be mutually exclusive, in the sense that responses that have minimal effects upon legitimate users may also have limited potential for safeguarding the system, and vice versa. For example, an attack against the confidentiality of the system could potentially be addressed by:

- delaying the disclosure of information, until an authentication challenge is issued;
- denying access to sensitive information by limiting permitted user behaviour;
- providing false information instead.

Each of these actions has different impact upon users and attacks. Delaying the disclosure of information, in order to issue an authentication challenge in the meantime, does not have significant impact upon the system, as the introduced delay could easily be disguised as usual system overload. However, the effect of the delay upon the attack is not significant either, as no action is taken to actually eliminate it. If the authentication challenge reinforces the suspicion of the system about the occurrence of an attack, it is possible to either deny access to the requested information, by limiting user behaviour, or provide false information instead. In both cases, the impact upon the system is the same, as the requested service is denied by the system. However the effect on the attack might be different, as the attacker who unsuccessfully attempts to compromise a system is likely to try again using another method. If false information is provided instead, the attacker is led to believe that the attack was successful, and thus the likelihood of attempting to break-in again is limited. That saves more time for the administrator and the IDS to counter any future attack, by patching vulnerabilities, increasing the monitoring level of the system, and developing defence mechanisms based on the security policy. Of course, false information provision could have significant adverse effects in a false positive scenario, as legitimate users could make decisions or act upon the false data. Thus the employment of such a method could be meaningful only for attacks with significant low false rejection rate.

It is clear that the issue of selecting appropriate responses to specific types of incident demands more structured analysis. It is necessary to consider the different classes of attacks, and their distinct characteristics that will influence the appropriateness of a response. To this end, the authors have designed a response-oriented taxonomy of IT system intrusions, which can be used as the basis for such analysis. Details of the taxonomy are presented in Papadaki et al (2002).

Assessment of influencing factors in intrusion response

There are numerous individual response actions that could be pursued in order to counter an intrusion, and some decision-making ability is required when a suspected incident presents itself. As previously identified, the AAIRS project has already conducted some research in this direction, identifying a number of factors that influence the response, along with the requirement for adaptive decision making. However, the authors consider that the range of contextual factors influencing response selection can be established in more detail than the AAIRS taxonomy has currently considered, and a number of further dimensions can be identified. This is illustrated in Figure 1, with the factors split according to whether they relate to the incident or the IDS.

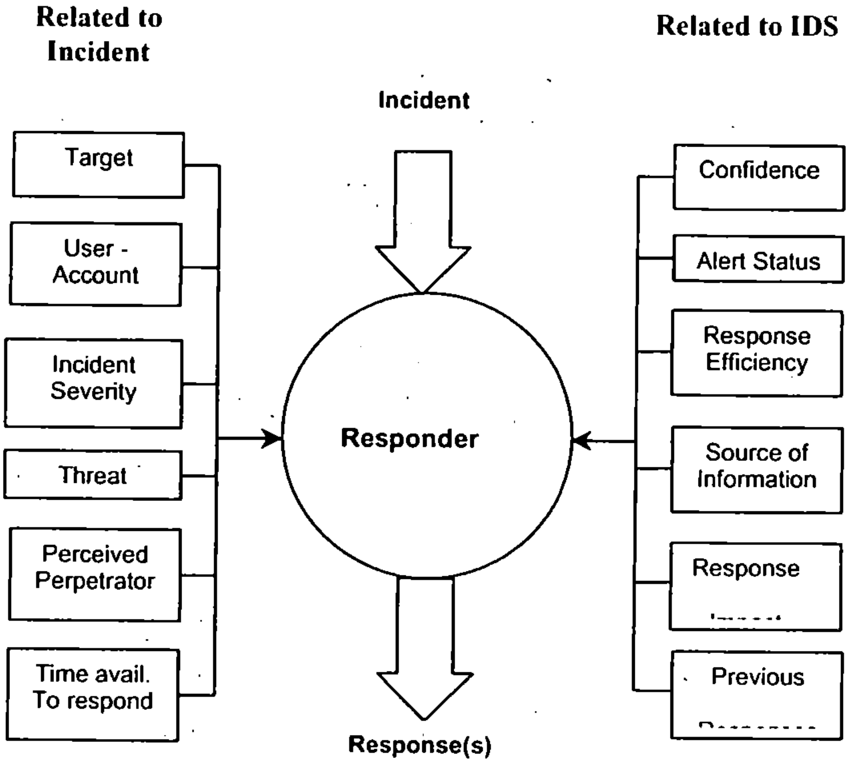


Figure 1: Contextual factors influencing intrusion response

As the figure shows, the *incident* is the trigger for the response and still represents the principal influence over what should be done. However, assessment of the other influencing factors enables the responder to establish the context in which the incident has occurred, and therefore select appropriate responses accordingly. The various factors are defined as follows:

Factors related to the incident

- **Target:** what system, resource or data appears to be the focus of the attack? What assets are at risk if the incident continues or is able to be repeated? How important is that resource for the continuation of the system operation?

- **User account:** if the attack is being conducted through the suspected compromise of a user account, what privileges are associated with that account? What risk do those privileges put the system on?

- **Incident severity:** what impact has the incident already had upon the confidentiality, integrity or availability of the system and its data? How strong a response is required at this stage? For example, the detection of a severe incident could warrant the initiation of correspondingly severe responses, in order to protect system resources.

- **Threat posed by incident:** how serious is the threat to the system, after the occurrence of the incident? Which attacks are more likely to follow, after that incident?

- **Perceived perpetrator:** does the evidence collected suggest that the perpetrator is an external party or an insider? Is there any history associated with that person/account?

- **Time available to respond:** How urgently is a response needed? This factor will be influenced by several of the other factors.

- **Factors related to the IDS**

- **Confidence:** how many monitored characteristics within the system are suggestive of an intrusion having occurred?

- **Alert status:** what is the current status of the monitoring system, both on the suspect account / process and in the system overall?

- **Response efficiency:** what has the efficiency of a specific response proven to be under specific conditions? The IDS will gradually update the efficiency rating of a specific response, after considering its efficiency in previous incidents. For example, for some types of attacks, targets, or attackers, some responses might be more efficient than others.

- **Source of Information:** what is the detecting capability of the source of information about the incident? Some sources or IDS metrics might be more efficient in detecting attacks than others, generating less false positive alarms (e.g. anomaly detectors tend to generate more false positive alarms than misuse detectors (Bace and Bell 2001), and some monitoring sensors produce less false alarms than others, depending on their location and configuration). The IDS should be able to determine the credibility of sources over time and adjust the confidence of the system on the probability of an intrusion.

- **Response impact:** what would be the impact of initiating a particular form of response? How would it affect a legitimate user if the suspected intrusion were, in fact a false alarm? Would there be any adverse impacts upon other system users if a particular response action were taken? Would it be possible to eliminate any adverse impacts and return the system to its initial state?

- **Previous Responses:** have any responses already been issued as a result of this incident? If one or more responses have already been issued and been unsuccessful in countering the intrusion, it would be relevant to consider this before determining the acceptable impact of the next action. The failure of previously issued responses might lead to the selection of more severe response actions (or an increase of the overall alert status of the system).

At the heart of Figure 1 was an entity referred to as the *Responder*. This is the element that will assess the various factors in order to select and invoke the required response(s). In current systems, this role is most likely to be fulfilled by a system administrator. However, in the context of an automated approach, the role would be assumed by a software-based agent, which itself would be an element of a wider intrusion monitoring system (Furnell and Dowland 2000).

Although Figure 1 highlights a number of factors, more thought should be given to the effect of different factors on the response decision mechanism. Indeed, identifying the factors that can influence response is only one part of the problem. The way in which one factor can influence others (i.e. the interrelationships between them) must also be analysed in order to determine the mechanism of the responder. For example, the type of target can influence the severity of the incident, as the more important the targeted system is (or the more vulnerable it is to specific attacks) the more severe the incident can become. In its turn, the severity of an intrusion can also influence several factors, such as the urgency to respond, and the acceptable impact of response (the more severe an intrusion is, the less transparent the response can be). All such ways in which factors can influence one another need to be identified and analysed in order to proceed with the conceptual design of the response framework.

Other outstanding issues at this stage include the relative weightings that should be assigned to the different factors in the response decision-making process. Some factors are likely to exert more significant influence than others, and the modelling of inter-relationships will clarify this to some extent. In addition, however, it is anticipated that weightings may alter according to the type of incident involved.

A final, yet crucial, aspect that requires investigation is the extent to which the various contextual factors can actually be measured in practice. Whilst all of them make sense at a conceptual level, obtaining the necessary information to quantify them in an operational system may be non-trivial.

Conclusion

The paper has established the importance of intrusion response within the context of IDS systems. Although the concept is represented to some extent within current systems, the most prevalent approaches are of a passive nature, aiming only to notify other parties about the occurrence of an attack, and then relying on them to take appropriate action. Automated active responses have the potential to offer a greater level of protection, since they can include actions to actually counter attacks. However, fears are currently expressed in the security community that automated responses introduce the danger of causing negative effects on a system, in case of a false positive alarm scenario. Nonetheless, an automated capability is desirable in that it will ease the administrative workload, and can protect systems from automated attack tools around the clock.

A broadening of response methods is necessary to extend the possibilities beyond the largely passive options that exist at present. Where possible, responses must be identified that have the potential for maximum impact upon an intruder, whilst minimising the effects upon legitimate users.

In order to increase confidence in the ability of automated response systems, the decision making process that underpins the selection of responses must be enhanced. This paper has summarised a range of factors that can influence the decision process. However, a deeper level of analysis is required in order to determine the relative importance, and consequent weightings, of factors in different scenarios, as well as potential inter-relationships between them. These aspects represent ongoing elements of research, and further findings will be documented in later publications.

References

Allen J., Christie A., Fithen W., McHugh J., Pickel J., Stoner E. (2000) *State of the Practice of Intrusion Detection Technologies*, Carnegie Mellon University, Technical Report CMU/SEI-99-TR-028, URL:
http://www.sei.cmu.edu/publications/documents/99.reports/99tr028/99tr028abstract.html

Amoroso E. (1999) *Intrusion Detection: An Introduction to Internet Surveillance, Correlation, Traps, Trace Back, and Response*, Second Printing, Intrusion.Net Books, New Jersey, June 1999.

Bace R., Mell P. (2001) *NIST Special Publication on Intrusion Detection Systems*, National Institute of Standards and Technology (NIST), URL:
http://csrc.nist.gov/publications/nistpubs/800-31/sp800-31.pdf

Carver C.A.Jr., Hill J.M.D., Surdu J.R., Pooch U.W (2000) A Methodology for Using Intelligent Agents to provide Automated Intrusion Response, *IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop*, West Point, New York, June 6-7 2000.

Carver C.A.Jr., Pooch U.W. (2000) An Intrusion Response Taxonomy and its Role in Automatic Intrusion Response, *IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop*, West Point, New York, June 6-7 2000.

Cheung S., Levitt K.N. (1997), Protecting Routing Infrastructures from Denial of Service Using Cooperative Intrusion Detection, *Proceedings of the New Security Paradigms Workshop*, Langdale,Cumbria UK, September 23 - 26, 1997, URL:
http://riss.keris.or.kr:8080/pubs/contents/proceedings/commsec/283699/

Cisco Systems (1998), *Cisco Secure Intrusion Detection System: Technical Overview*, Cisco Systems Inc., URL: http://www.cisco.com/warp/public/cc/pd/sqsw/sqidsz/tech/ntran_tc.htm

Cohen F.B. (1999), Simulating Cyber Attacks, Defenses, and Consequences, *Infosec Baseline Studies*, URL: http://all.net/journal/ntb/simulate/simulate.html

Computer Associates (1999), *White Paper – eTrust Intrusion Detection*, September 6, 1999: http://www.cai.com/solutions/enterprise/etrust/intrusion_detection/product_info/sw3_whitepaper.htm

Computer Associates (2001) *SessionWall3 – Intrusion Detection*,:
http://ca.com/solutions/enterprise/etrust/sw_intrusion_detection/product_info/sw3_intrusion.htm

Denning D.E. (1987), An Intrusion-Detection Model, *IEEE Transactions on Software Engineering*, Vol. **SE-13**, 2 :222-232.

Furnell S.M., Dowland P.S. (2000) A conceptual architecture for real-time intrusion monitoring , *Information Management & Computer Security*, **8**, 2:65-74.

Gilliom G., Proctor P. E. (2001) *The Case for Centrac ICE Hybrid Security Solution*, NetworkICE Corporation – CyberSafe Corporation, March 2001: http://www.centraxice.com/centrax/content/CentraxICE_whitepaper.pdf/

Hurwitz Group (2000) *Hurwitz Report: AXENT Technologies' NetProwler™ and Intruder Alert™*, Hurwitz Group Inc., September 2000. URL: http://www.hurwitz.com/

Internet Security Systems (2001) *RealSecure 6.0: Frequently Asked Questions*, June 14, 2001, pp. 5-6. URL: http://documents.iss.net/literature/RealSecure/rs60_faq.pdf

Lee S.Y.J. (2001) *Methods of response to IT system intrusions*, MSc thesis, University of Plymouth, Plymouth, UK, September 2001.

Mukherjee B., Heberlein L.T.; Levitt K.N. (1994) Network Intrusion Detection, *IEEE Networks*, **8**,3:26-41.

Papadaki, M., Furnell, S.M., Lines, B.M., Reynolds, P.L. (2002) A response-oriented taxonomy of IT system intrusions, in M.Roccetti (ed.), *Proceedings of EUROMEDIA 2002*, Modena, Italy, April 2002. pp:87-95.

Porras P.A., Neumann P.G. (1997), EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances, *20th National Information Systems Security Conference*, October 9, 1997. URL: http://www.sdl.sri.com/projects/emerald/emerald-niss97.html

Power R. (2001) 2001 CSI/FBI Computer Crime and Security Survey, *Computer Security Issues and Trends*, **VII**, 1.

Ragsdale J.D., Carver C.A. Jr., Humphries J.W., Pooch U.W. (2001) Adaptation Techniques for Intrusion Detection and Intrusion Response Systems, *2nd Annual IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop*, West Point New York, June 5-6, 2001.

Schneier B. (2000) Secrets and Lies: Digital Security in a Networked World, John Wiley & Sons, New York.

Shipley G. (1999), Intrusion Detection, Take Two, *Network Computing online journal*, November 15, 1999. URL: http://www.nwc.com/

SRI (2000) *EMERALD expert-BSM: Capabilities*, SRI International. URL: http://www.sdl.sri.com/projects/emerald/releases/eXpert-BSM/cap.html

UC Davis (2000) *UC Davis Response and Detection Project Overview*, December 2000, URL: http://seclab.cs.ucdavis.edu/response/

Operational Characteristics of an Automated Intrusion Response System

Maria Papadaki[1], Steven Furnell[1], Benn Lines[1], and Paul Reynolds[2]

[1] Network Research Group, University of Plymouth,
Drake Circus, Plymouth, United Kingdom
info@network-research-group.org
[2] Orange Personal Communications Services Ltd,
St James Court, Great Park Road, Bradley Stoke,
Bristol, United Kingdom.

Abstract. Continuing organisational dependence upon computing and networked systems, in conjunction with the mounting problems of security breaches and attacks, has served to make intrusion detection systems an increasingly common, and even essential, security countermeasure. However, whereas detection technologies have received extensive research focus for over fifteen years, the issue of intrusion response has received relatively little attention - particularly in the context of automated and active response systems. This paper considers the importance of intrusion response, and discusses the operational characteristics required of a flexible, automated responder agent within an intrusion monitoring architecture. This discussion is supported by details of a prototype implementation, based on the architecture described, which demonstrates how response policies and alerts can be managed in a practical context.

1 Introduction

Ever since the commercialisation of the Internet, there has been a substantial growth in the problem of intrusions, such as Denial of Service attacks, website defacements and virus infections [1]. Such intrusions cost organisations significant amounts of money each year; for example, the 2003 CSI/FBI Computer Crime and Security Survey [2] reported annual losses of $201,797,340 from 530 companies questioned. Although these results suggest that the cost of attacks has decreased for the first time since 1999, it is still significant amount, representing a 101.55% increase compared to 1997 [3].

As a defence against such attacks, intrusion detection technologies have been employed to monitor events occurring in computer systems and networks. Intrusion detection has been an active research area for more than 15 years [4,5], and merits a wide acceptance within the IT community [6;3]. However, detecting intrusions is only the first step in combating computer attacks. The next step involves the counteraction of an incident and has so far been largely overlooked [7;8]. The CSI/FBI survey suggests a declining trend amongst organisations to address vulnerabilities, or report incidents to law enforcement since 1999 [2]. Although the percentage of respondents,

A. Lioy and D. Mazzocchi (Eds.): CMS 2003, LNCS 2828, pp. 65-75, 2003.
© IFIP International Federation for Information Processing 2003

who patched vulnerabilities after an incident, was reasonably high, it was still decreased by 2% when compared to the respective figure of 1999, while about 50% of the respondents chose not to report the incident at all. Even if vulnerability patching and incident reporting are only two aspects of responding to intrusions, the lower percentages suggest a lack of effective response policies and mechanisms within organisations.

A principal reason for this problem is likely to be the administrative overhead posed by response procedures. At the moment, the detection of a suspected intrusion typically triggers a manual intervention by a system administrator, after having received an alert message from the intrusion detection system. The IDS can additionally assist the incident response process, by providing the details of the attack, saved in a log file [9]. However, responding manually to intrusions is not necessarily an easy task, as it may involve dealing with a high number of alerts and notifications from the IDS [10], ensuring awareness of security bulletins and advisories from incident response teams, and taking appropriate actions to resolve each of the alerts reported. From the system administrator's perspective, the main requirement is to ensure that the system remains operational and available. Thus, unless resolving a detected incident is explicitly required to ensure that this is the case, the task of responding is likely to be given a lower priority.

The importance of timely response has been demonstrated by Cohen [11] in his simulation of attacks, defences and their consequences in complex 'cyber' systems. These showed that, if skilled attackers are given 10 hours between being detected, and generating a response, then they have an 80% chance of a successful attack. When that time interval increases to 20 hours, the rate of success rises to 95%. After 30 hours the skill of the system administrator makes no difference, as the attacker will always succeed. However, if the response is instant, the probability of a successful attack against a skilled system administrator becomes almost zero. This shows not only the importance of response, but also the relationship between its effectiveness and the time it is initiated.

At the time of writing, the degree of automation in current IDS is very low, offering mostly passive responses (i.e. actions that aim to notify other parties about the occurrence of an incident and relying on them to take further action). In contrast, active responses (actions taken to counter the incident that has occurred) either have to be initiated manually or may not be offered at all. Lee [12] found that even if IDS products offer active responses, they are not trusted by administrators, mainly due to the likely adverse effects in the event of them being falsely initiated. In spite of the potential problems, practical factors suggest that automated response methods will become increasingly important. For example, the widespread use of automated scripts to generate distributed attacks [13] can offer very limited opportunity to respond, and further diminishes the feasibility of doing so manually. Thus, there is a need for the adoption of automated response mechanisms, which will be able to protect system resources in real time and, if possible, without requiring explicit administrator involvement at the time.

As an effort to enhance the effectiveness of automated response and reduce its adverse effects in false rejection scenarios, an automated response framework has been devised. The aim is to enable accurate response decisions to be made autonomously, based on the nature of the attack and the context in which it is occurring (e.g. what applications are running, what account is being used, etc.). The

remainde.
a prototy

2 The

IMS has
years an
supervisic
a numbe
comparisc
behaviour
patterns.
and data
relevant t
modules c
readers ar
given to tl

The Re
Engine (n
papers) a
taking app
need to b
initiating
of actions
updating
increasing

The Re
decision.
Detection
possible c

As well
able to din
status of tl
the perceiv

The Co
system (e.
application
legitimate
important
actions. It
them need
software (
the respon
minimised

remainder of this paper describes the concept of the Responder, followed by details of a prototype implementation that demonstrates the approach in practice.

2 The Intrusion Monitoring System (IMS)

IMS has been the focus of research within the authors' research group for several years and is a conceptual architecture for intrusion monitoring and activity supervision, based around the concept of a centralised host handling the monitoring of a number of networked client systems. Intrusion detection is based upon the comparison of current user activity against both historical profiles of normal behaviour for legitimate users and intrusion specifications of recognised attack patterns. The architecture addresses data collection and response on the client side, and data analysis and recording at the host. The elements of the architecture that are relevant to the discussion presented in this paper are illustrated in Figure 1. The main modules of IMS have already been defined in earlier publications [14], and interested readers are referred to these for associated details. In this paper, specific focus will be given to the modules related to intrusion response.

The Responder is responsible for monitoring the Alerts sent from the Detection Engine (note: this module was referred to as the Anomaly Detector in previous papers) and, after considering them, in conjunction with other contextual factors, taking appropriate actions where necessary. If the actions selected by the Responder need to be performed on the client side, a local Responder Agent is responsible for initiating and managing the process. Without providing an exhaustive list, examples of actions that could be performed at the client side include correcting vulnerabilities, updating software, issuing authentication challenges, limiting access rights and increasing the monitoring level.

The Responder utilises a variety of information in order to make an appropriate decision. This is acquired from several other elements of IMS, including the Detection Engine, the Collector, the Profiles, and the Intrusion Specifications. The possible contributions from each of these sources are described below.

As well as indicating the type of suspected incident, the Detection Engine is also able to directly inform the Responder about the intrusion confidence, the current alert status of the IDS, the source of the alert that triggered the detection, information about the perceived perpetrator(s) and the target involved.

The Collector is able to provide information about current activity on the target system (e.g. applications currently running, network connections currently active, applications installed etc.). This information can be used to minimise the disruption of legitimate activity, by making sure that no important work at the target gets lost, or no important applications are ended unnecessarily, as a result of selected response actions. It can also be used for cases of compromised targets when information about them needs to be reassessed. For example, the determination of whether unauthorised software (sniffing software / malware) has been installed will be vital information for the response decision process. In that way the negative impacts of responses can be minimised and the response capability enhanced as much as possible.

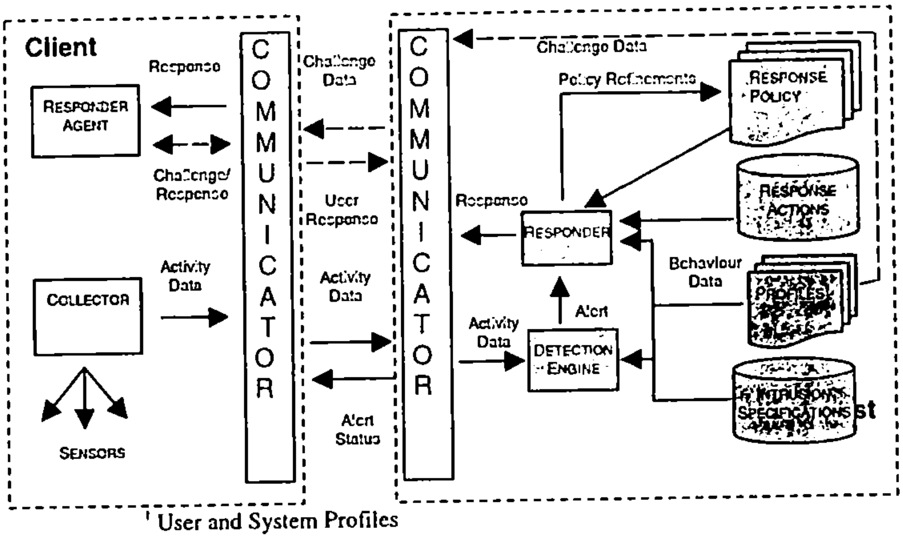


Fig. 1. The Intrusion Monitoring System (IMS)

The Profiles contain information about users and systems, both of which can provide some information in the context of response decisions:

- User profiles: If the incident involves the utilisation of a user account, then the corresponding user profile can indicate aspects such as the privileges and access rights associated with it.
- System profiles: These relate to system characteristics, such as versions of operating systems and installed services, the expected load at given hours/periods, the importance of the system within the organisation (e.g. whether it holds sensitive information or offers critical services), its location on the network etc.

Finally, Intrusion Specifications contain information about specific types of intrusions and their characteristics - such as incident severity rating, ratings of likely impacts (e.g. in terms of confidentiality, integrity and availability), and the speed with which the attack is likely to evolve [15]. Once the Detection Engine has indicated the type of incident that it believes to have occurred, additional information can be retrieved from the specifications to obtain a comprehensive view of the incident (all of which would again influence the response selection).

Having gathered all of the available information, the actions that should be initiated in different contexts are then specified in the Response Policy. In the first instance, the Response Policy would need to be explicitly defined by the system administrator; however, it could also be refined over time to reflect practical experience. For example, if a particular response is found to be ineffective against a particular situation, then the policy could be updated to account for this. It is envisaged that this refinement could be initiated manually by the system administrator, as well as automatically by the system itself. Further information about this process is given in the next section.

3 O

In ord
with
impro
curren

- ada
- ass

The c
the re
respo
occur
occas
under
draw
This
occur

- th
- wh
ha
- th
att
da
- th
at
W
- th
sy
- th
in
D
- th
H

Hav
cont
criti
sele
mos
ope
occu
indi
aler

3 Operational Characteristics of the Responder

In order to enable increasingly automated responses, and reduce the risks associated with using active response methods, the architecture incorporates techniques to improve the flexibility of the response process when compared to approaches in current IDS. Specifically, the proposed Responder includes the ability to:

- adapt decisions according to the current context; and
- assess the appropriateness of response actions before and after initiating them.

The concept of adaptive decision-making relates to the requirement for flexibility in the response process. A fundamental principle of the proposed approach is that response decisions should vary depending upon the context in which an incident has occurred (i.e. a response that is appropriate to a particular type of incident on one occasion will not necessarily be appropriate if the same incident was to occur again under different circumstances). The previous section described how the Responder draws upon information from a number of other sources within the IMS framework. This enables the system to determine the overall context in which an incident has occurred, including considerations such as:

- the overall alert status of the IDS at the time of the new incident;
- whether the incident is part of an ongoing series of attacks (e.g. how many targets have already been affected? Which responses have already been issued?);
- the perpetrator of the attack (is there enough information to suggest a specific attacker? Is he/she an insider/outsider? Has he/she initiated an attack before? How dangerous is he/she? What attacks is he likely to attempt?);
- the current status of the target (e.g. is it a business critical system? What is its load at the moment? Is there any information or service that needs to be protected? What software/hardware can be used for response?);
- the privileges of the user account involved (e.g. what is the risk of damage to the system?);
- the probability of a false alarm (how reliable has the sensor/source that detected the incident been in the past? What is the level of confidence indicated by the Detection Engine about the occurrence of an intrusion?);
- the probability of a wrong decision (how effective has the Responder been so far? Have these responses been applied before in similar circumstances?).

Having assessed the above factors, response decisions must then be adapted to the context accordingly. For example, if the incident has been detected on a business critical system, and the Detection Engine has indicated a low confidence, then the selection of a response with minimal impact upon the system would represent the most sensible course of action. That decision minimises the chance of critical operations being disrupted in the case of an error alert. However, if the same scenario occurred in conjunction with previous alerts having already been raised (i.e. indicating that the current incident was part of a series of attacks), or if the overall alert status of the IDS was already high, then a more severe response would be warranted. More comprehensive information about this decision process, and the information that would be assessed, is presented in earlier publications [15; 16].

The other novel feature of the Responder is its ability to assess the appropriateness of response actions. This can be achieved in two ways; firstly by considering the potential side effects of a response action, and secondly by determining its practical effectiveness in containing or combating attacks.

As previously identified in the introduction, the problem of side effects is a particular concern in the context of using active responses, because they have the potential to adversely affect legitimate users of the system. As a result, this needs to be considered before the Responder chooses to initiate a given action. There are a number of characteristics that would be relevant in this context:

- the transparency of the response action. In some cases it might be preferable to issue responses that do not alert the attacker to the fact that he/she has been noticed, whereas in others it could be preferable to issue a response that is very explicit.
- the degree to which the action would disrupt the user to whom it is issued. This is especially relevant in the context of a response action having been mistakenly issued against a legitimate user instead of an attacker. In situations where the Detection Engine has flagged an incident but expressed low confidence, it would be desirable to begin by issuing responses that a legitimate user would be able to overcome easily.
- the degree to which the action would disrupt other users, or the operation of the system in general. Certain types of response (e.g. termination of a process, restriction of network connectivity) would have the potential to affect more than just the perceived attacker, and could cause reduced availability to other people as well. As such, the Response Policy may wish to reserve such responses only for the most extreme conditions.

Each of these factors would need to be rated independently, and the information would be held in the database of available response actions (previously illustrated in Figure 1). The consideration of the ratings could then be incorporated into the response selection process as appropriate, and indeed during the formulation of the Response Policy by the system administrator. In addition to assessing the side effects, each response could also usefully be given an associated rating to indicate its perceived strength (which could inform the Responder and the administrator about its likely 'stopping power' in relation to an attacker).

The second factor that would influence the appropriateness of a response in a particular context would be whether it had been used in the same context before. If the Responder keeps track of its previous response decisions, then they can subsequently be used as the basis for assessing whether the response actions were actually effective or not. This requires some form of feedback mechanism, which can then be used to refine the Response Policy. It is envisaged that feedback could be provided in two ways: explicitly by a system administrator, and implicitly by the Responder itself. In the former case, the administrator would inspect the alert history and manually provide feedback in relation to the responses that had been selected to indicate whether or not they had been effective or appropriate to the incident. By contrast, the latter case would require the Responder itself to infer whether previous responses had been effective. A simplified example of how it might do this would be to determine whether it had been required to issue repeated responses in relation to the same detected incident. If this was the case, then it could potentially infer that (a) the initial response actions were not effective against that type of incident, and (b) the last response action issued might form a better starting point on future occasions (i.e. upgrading and downgrading the perceived effectiveness of the responses when used in that context).

Having obtained such feedback, it would be desirable for the system to automatically incorporate it into a refined version of the Response Policy. This, however, would be a non-trivial undertaking, and it is anticipated that a full implementation of the system would need to incorporate machine-learning mechanisms to facilitate a fully automated process. An alternative would be to collate the feedback, and present it to the system administrator for later consideration when performing a manual overhaul of the Response Policy.

4 A Prototype Responder System

As an initial step towards the development of the Responder, a prototype system has been implemented that demonstrates the main response features of IMS, including the ability to make decisions based on the information from IDS alerts and other contextual factors.

The first element of the prototype is a console used to simulate intrusion conditions. In the absence of a full Detection Engine, or indeed genuine incidents, this is necessary to enable incident conditions to be configured before generating an alert to trigger the Responder's involvement. The parameters that can be adjusted from the console interface include the ones that are meant to be provided by the Detection Engine in the alert message, and are illustrated in Figure 2. The Responder can form a decision by monitoring (or determining) an additional set of contextual parameters, and then using these in conjunction with the ones included in the alert message.

The second component of the prototype is the Responder itself, which is responsible for receiving the alerts and making response decisions according to the given context. The Responder largely bases its decision upon the Response Policy, which can be accessed from the Responder module, by selecting the Response Policy Manager tool. A user-friendly interface is provided for the review of Policy rules, which are represented via a hierarchical tree, where the incidents are at the highest level and the response actions lie at the lowest levels. At the most basic level, there will be a one to one correspondence between a type of incident and an associated type of response. However, a more likely situation is that the desired response(s) to an incident will vary, depending upon other contextual factors, and the Policy Manager allows these alternative paths to be specified via intermediate branches in the tree. Between them, these intermediate branches comprise the conditions, under which specific response actions are initiated for particular incidents.

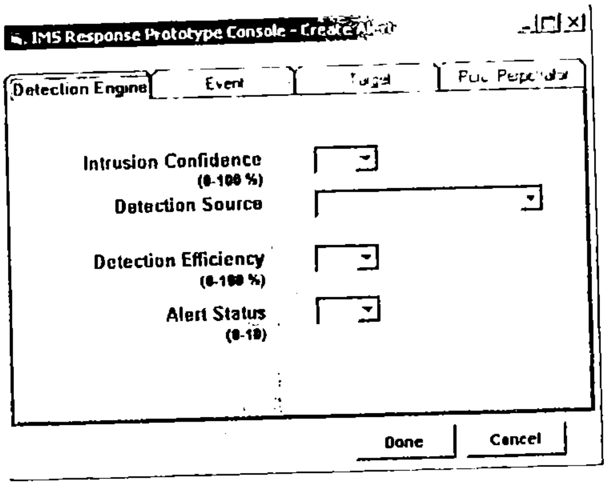
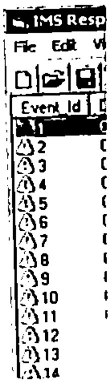


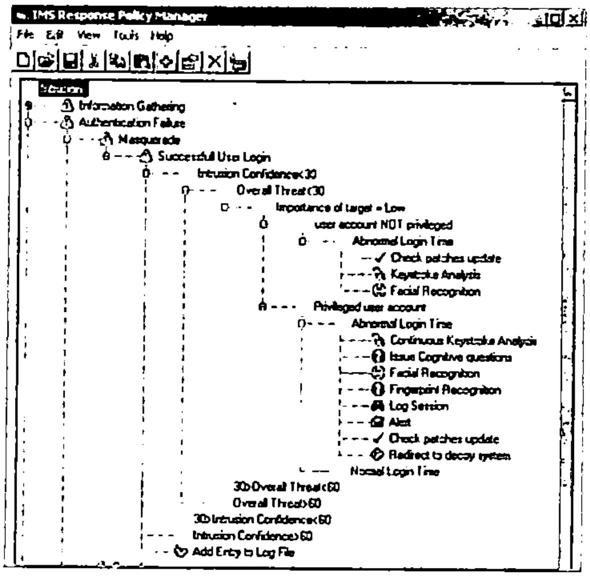
Fig. 2. Prototype Console Interface

The IMS Response Policy Manager is illustrated in Figure 3, with an example of response rules that could be specified in relation to an 'authentication failure' incident. In this case, had there been an alarm from the Detection Engine describing the successful login of a suspected masquerador, the Responder would check for the most recent update of related software to ensure that it is not vulnerable, and initiate keystroke analysis and facial recognition (if available) to authenticate the user in a non-intrusive manner. Of course, the conditions for the latter to happen would not be just the occurrence of the incident. Only the addition of the alarm to a log file would happen in that case. For the previously mentioned responses to be issued, the intrusion confidence would need to be low (hence the responder would need to collect more information about the incident), the overall threat and the importance of the target would need to be at low levels as well, not justifying the issue of more severe responses. Also, the account involved would need to be not privileged, with login time outside the normal pattern, in order to issue non-intrusive authentication.

Had there been a privileged account logged in at an abnormal time, then the urgency to collect more information about the incident would be greater and thus more intrusive countermeasures could be allowed. More authentication challenges like continuous keystroke analysis [17], the use of cognitive questions [18], and fingerprint recognition could also be used. Other methods that could be utilised include session logging (for further future reference or forensic purposes), alerting the user himself/herself about the occurrence of this suspicious behaviour (aiming to provoke a reaction from him/her and possible discourage him/her from any further unauthorised activity). Finally another option would be the redirection to a decoy system, in order to protect the integrity of the original target. Although this option would be more suited in the case of a server being compromised, it could still be an option for very sensitive environments, where a maximum level of security is required and minimum levels of risk are allowed. In any case, Figure 3 depicts an example of a security policy, which may or may not be optimal.

Having dete
alerts it re
responses tl
administrat
contains a
response a
incident its
the alerts, i
was made l
selected. A
for automa

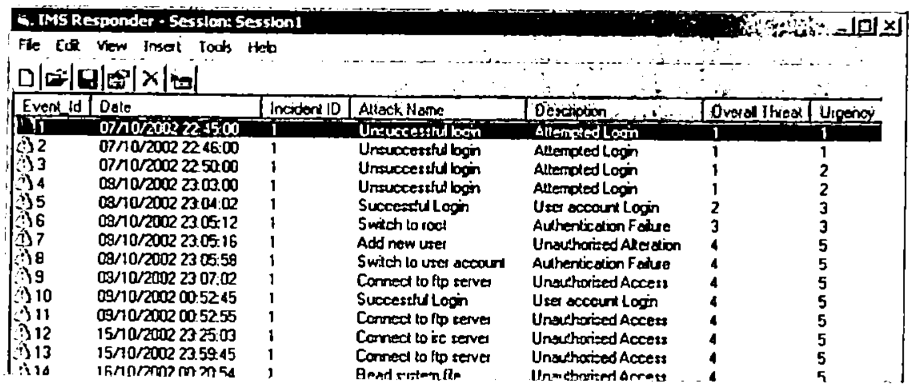




- Information Gathering
- Authentication Failure
  - Masquerade
    - Successful User Login
      - Intrusion Confidence<30
        - Overall Threat<30
          - Importance of target = Low
            - user account NOT privileged
              - Abnormal Login Time
                - Check patches update
                - Keystroke Analysis
                - Facial Recognition
            - Privileged user account
              - Abnormal Login Time
                - Continuous Keystroke Analysis
                - Issue Cognitive questions
                - Facial Recognition
                - Fingerprint Recognition
                - Log Session
                - Alert
                - Check patches update
                - Redirect to decoy system
              - Normal Login Time
        - 30>Overall Threat<60
        - Overall Threat>60
      - 30>Intrusion Confidence<60
      - Intrusion Confidence>60
      - Add Entry to Log File

Fig. 3. IMS Response Policy Manager

Having determined the Response Policy, the Responder can make decisions about the alerts it receives. During normal operation, the Responder logs the details of responses that have been issued so that they can be tracked and reviewed by a system administrator. This is achieved via the Alert Manager interface (see Figure 4), which contains a list of suspected incidents, allowing them to be selected and reveal the response action(s) initiated for them. Each alert contains information about the incident itself, and the reasoning for the associated response decision. When viewing the alerts, it is also possible for the administrator to review the response decision that was made by the system, and provide feedback about the effectiveness of the actions selected. A full implementation of the Responder would use this feedback as the basis for automatic refinement of the response policy over time.


IMS Responder - Session: Session1
File Edit View Insert Tools Help

Event_Id	Date	Incident ID	Attack Name	Description	Overall Threat	Urgency
1	07/10/2002 22:45:00	1	Unsuccessful login	Attempted Login	1	1
2	07/10/2002 22:46:00	1	Unsuccessful login	Attempted Login	1	1
3	07/10/2002 22:50:00	1	Unsuccessful login	Attempted Login	1	2
4	08/10/2002 23:03:00	1	Unsuccessful login	Attempted Login	1	2
5	08/10/2002 23:04:02	1	Successful Login	User account Login	2	3
6	08/10/2002 23:05:12	1	Switch to root	Authentication Failure	3	3
7	08/10/2002 23:05:16	1	Add new user	Unauthorized Alteration	4	5
8	08/10/2002 23:05:58	1	Switch to user account	Authentication Failure	4	5
9	08/10/2002 23:07:02	1	Connect to ftp server	Unauthorised Access	4	5
10	09/10/2002 00:52:45	1	Successful Login	User account Login	4	5
11	09/10/2002 00:52:55	1	Connect to ftp server	Unauthorised Access	4	5
12	15/10/2002 23:25:03	1	Connect to irc server	Unauthorised Access	4	5
13	15/10/2002 23:59:45	1	Connect to ftp server	Unauthorised Access	4	5
14	16/10/2002 00:20:54	1	Read system file	Unauthorised Access	4	5

Fig. 4. IMS Responder: Alert Manager

5 Conclusions and Future Work

This paper has presented the requirements for enhanced intrusion response and the operational characteristics of an automated response architecture that enables flexible, escalating response strategies. The prototype system developed provides a proof-of-concept, and demonstrates the process of creating and managing a flexible response policy, as well as allowing intrusion scenarios to be simulated in order to test the response actions that would be initiated. Although the IMS approach as a whole would not necessarily be suited to all computing environments it is considered that the automated response concept could still be more generally applicable.

Future work could usefully include the integration of machine learning algorithms into the Responder implementation, in order to enable it to learn from the effectiveness (or otherwise) of previous response decisions and automatically refine the response policy accordingly. Based on the feedback from experience, the ability to learn and to assess its decision-making capability, the Responder could eventually attain a sufficient level of confidence to operate autonomously.

Acknowledgments. The research presented in this paper has been supported by funding from the State Scholarships Foundation (SSF) of Greece.

References

1. CERT Coordination Center: Security of the Internet, Vol. 15, The Froehlich/Kent Encyclopedia of Telecommunications, Marcel Dekker, New York (1997) 231–255
2. Richardson, R.: 2003 CSI/FBI Computer Crime and Security Survey (2003) http://www.gocsi.com/
3. Power, R.: 2002 CSI/FBI Computer Crime and Security Survey, Vol. VIII, No. 1, Computer Security Issues and Trends (2002) 10–11, 20–21
4. Denning, D.E.: An Intrusion-Detection Model, Vol. SE-13, No. 2, IEEE Transactions on Software Engineering (1987) 222–232
5. Allen, J., Christie, A., et al.: State of the Practice of Intrusion Detection Technologies, Technical Report CMU/SEI-99-TR-028, Carnegie Mellon University (2000) http://www.sei.cmu.edu/publications/documents/99.reports/99tr028/99tr028abstract.html
6. Mukherjee, B., Heberlein, L.T.; Levitt, K.N.: Network Intrusion Detection, *IEEE Networks 8*, no.3 (1994) 26–41
7. Schneier, B.: Secrets and Lies: Digital Security in a Networked World, John Wiley & Sons (2000)
8. Amoroso, E.: Intrusion Detection: An Introduction to Internet Surveillance, Correlation, Traps, Trace Back, and Response, Second Printing, Intrusion.Net Books, New Jersey (1999)
9. Bace, R., and Mell, P.: NIST Special Publication on Intrusion Detection Systems, National Institute of Standards and Technology (NIST), http://csrc.nist.gov/publications/drafts/idsdraft.pdf (2001)

10. Newman, D., Snyder, J., and Thayer, R.: Crying Wolf: False Alarms hide attacks, Network World Fusion Magazine, http://www.nwfusion.com/techinsider/2002/0624security1.html/ (2002)
11. Cohen, F.B.: Simulating Cyber Attacks, Defences, and Consequences, The Infosec Technical Baseline studies, http://all.net/journal/ntb/simulate/simulate.html (1999)
12. Lee, S.Y.J.: Methods of response to IT system intrusions, MSc thesis, University of Plymouth, Plymouth (2001)
13. Cheung, S., and Levitt, K.N.: Protecting Routing Infrastructures from Denial of Service Using Cooperative Intrusion Detection, Proceedings of the New Security Paradigms Workshop, Langdale,Cumbria UK (1997)
http://riss.keris.or.kr:8080/pubs/contents/proceedings/commsec/283699/
14. Furnell, S.M., and Dowland, P.S.: A conceptual architecture for real-time intrusion monitoring, Vol. 8, No. 2, Information Management & Computer Security (2000) 65-74
15. Papadaki, M., Furnell, S.M., Lines, B.M., and Reynolds, P.L.: A Response-Oriented Taxonomy of IT System Intrusions, Proceedings of Euromedia 2002, Modena, Italy (2002) 87–95
16. Papadaki, M., Furnell, S.M., Lee, S.J, Lines, B.M., and Reynolds, P.L.: Enhancing response in intrusion detection systems, Vol. 2, No. 1, Journal of Information Warfare (2002) 90–102
17. Dowland, P., Furnell, S., and Papadaki, M.: Keystroke Analysis as a Method of Advanced User Authentication and Response, Proceedings of IFIP/SEC 2002 - 17th International Conference on Information Security, Cairo, Egypt (2002) 215–226
18. Irakleous, I., Furnell, S., Dowland, P., and Papadaki, M.: An experimental comparison of secret-based user authentication technologies, Vol. 10, No. 3, Journal of Information Management & Computer Security (2002) 100–108

can execute active content, which in turn may be able to make calls and send multimedia messages. If active content is able to create active content to other devices then, of course, self-replication is possible and thus viruses infecting mobile devices are possible. Until now, mobiles phones have been closed environments, but that is changing. We do not know what will happen after the Cabir virus, but what we do know is that current mobile technology allows viruses to exist in mobile devices. What is frightening is that mobile phones are likely to have more and more complexity, features, and capacity."

Now that we know that these kinds of levices can be infected by malicious code vhat should we do? Leave Bluetooth disibled unless you really need it because ıpart from the mobile phone virus, which s not in the wild, Bluesnarfing has ırrived. "For Bluesnarfing to be successul", says Colm Murphy, "the sender and he recipients need Bluetooth enabled on heir mobile phones. You simply search or all the Bluetooth enabled devices vithin a 30-feet radius and send your nessage. This could be a derogatory emark, a marketing ploy at an Expo— ree coffee and cakes at stand 51', or anything your imagination feels free to conjure up. Things could easily get out of hand with this facility, especially in relation to bullying or sexual harassment. One way to avoid this and take yourself out of the loop is to set your phone to only accept or send messages from and to a preferred list, or simply disable Bluetooth", says Murphy.

"I think the 'Expo' example is a good one", says Murphy. "It is perfect ground to release a mobile virus that spreads quickly and targets a specific audience. Or what about a mobile phone virus that makes the phone dial a specific number! Some disgrunded ex-employees with a grudge could have some fun with that one!"

"The big concern is that in the future large mobile virus outbreaks may be reality", says Dr. Helenius. "Denial-of-service attacks may affect critical infrastructures, like emergency phone numbers. Indeed, an efficient virus may be able to block phone lines and phone networks, like an efficient Internet worm can block Internet connections. However, we do not know if such disasters will happen. The mobile device and network developers have a choice. They can adapt more security in their products in order to prevent disasters. Software could be written securely in order to prevent buffer overflows and other critical errors. More importantly, security can be an essential part of design. For example, it is possible to adapt hardware components that will prevent unauthorized phone calls. We should not merely trust software, and security should be based on more than one layer. If one security layer fails, there could be other security layers that will prevent further damage." On a more skeptical note Colm Murphy says, "If you consider the tens of millions of EURO that companies are investing in developing products that stop these kinds of threats, I think it is safe to assume that we will see more mobile phone viruses as time passes. The investors will demand it!!!"

References:

[1] It spreads to devices that run under Symbian OS, which is used in many models of phones manufactured by Nokia, Siemens, Sony and Ericcson.

[2] Even a Bluetooth-enabled printer according to the Symantec security response.

[3] (www.symantec.com)

DS or IPS: what is best?

Maria Papadaki and Steven Furnell University of Plymouth,

ıtrusion detection systems (IDS) have become one of the most common counrmeasures in the network security arsenal. But while other technologies such as rewalls and anti-virus provide proactive protection, most current IDSs are pasve; detection of a suspected intrusion typically triggers a manual response from system administrator. Too often, this comes too late.

ohen has demonstrated the importance quick response. In a simulation study showed that if 10 hours elapse between tection and response, then attackers ve an 80% chance of success. At 20 urs, the success rate rises to 95%, and er 30 hours the attacker will always cceed, regardless of the skill of the ministrator. But if the response is tant, the probability of success against killed administrator is almost zero. [1]

Such findings are valuable in view of the rapid escalation that characterises many of today's Internet-based attacks. Recent incidents such as Sasser and MyDoom have shown us that we do not have the luxury of time to react.

Such factors have led to increased interest in an alternative technology, namely the intrusion prevention system (IPS). Although these incorporate intrusion detection mechanisms, and share similarities such as being deployable in both network and host-based contexts, they also have two significant differences. Instead of passively monitoring activity on systems and networks, IPSs are positioned inline and can therefore block unauthorized activity before it takes place (see Figure 1). In a network context, conceptually they combine firewall approaches with intrusion detection capabilities; in host environments, they monitor all system and API calls and block those that would cause malicious behaviour. [2]

With reference to Figure 1, products are now available that can be configured to operate in either mode (an example being McAfee's Intrushield). However, as later discussion will establish, this is-not to suggest that the use of IDS and IPS is an either/or decision.

The IDS is dead, long live the IPS?

Although IPS solutions have been available for several years, their adoption had been limited. More recently, however, there has been a shift in the attitudes of vendors and users in relation to IDS, and in the characteristics of the products on offer.

A notable contributor to this was a market report from Gartner in June 2003. This set tongues wagging because it branded IDS technology a "market failure", and predicted that it would be dead by 2005.[3] The report suggested that customers hold off big investments in IDS because the technologies added no practical value in enterprise security. Reaction to this report from IDS vendors and security specialists was intense.[4]

Gartner argued against IDS mainly because of their inability to prevent intrusions, and the vast number of false positive alarms they can generate. False alarms are indeed a recognized problem with IDS, and are the bane of many security administrators' lives. A 2003 survey by OpenService, Inc (www.open.com) established that management of false positives is among the top three problems facing security practitioners; only shrinking budgets and threat risk assessments raised more concern. [5]

The tendency of IDS to generate false positives also has the undesirable side-effect that administrators tire of following-up dead ends and become slack about tracking fresh alerts.

Gartner's other main point, that IDS does not prevent intrusions, is also fair. Usually this is because the IDS is placed out of band as a monitoring device, with its response capability restricted to passive actions such as logging data and issuing alerts. Given the problem of false positives, it is understandable that IDS is not often trusted to respond more actively, such as blocking traffic, ending sessions, restricting access and the like.

The debate had forced many IDS vendors to incorporate intrusion prevention solutions in their products. Even where vendors have not adopted preventi on solutions, the term "intrusion detection system" tends to be avoided. In its place people talk of "intrusion management system" or "intrusion protection system". This may be to distance products from any doubts in potential customers' minds. Indeed, some effects amongst the user community can also be observed. For example, for the first time, the CSI/FBI security survey reports fewer respondents using IDS technology (see Figure 2). It is also notable that the 2004 survey was the first to ask respondents specifically about the use of IPS technology. The question got a 45% response rate.[6]

So, why the sudden interest in IPS products? Is it vendors running scared, wanting to distance themselves from the fallout from the Gartner report? Is it a marketing exercise? Refocusing of products certainly has the potential to press the right buttons from a consumer perspective—after all, why would you want to buy a detection product if you can get one that actually prevents intrusions? Or have we hit upon a technology that solves the problem of attacks, without the perceived weaknesses of IPS? In short . . .

Is IPS really an alternative?

The ability to stop intrusions logically suggests a maturing of the technology. It suggests that intrusion detection technologies have become accurate enough for us to rely upon their decisions to be correct. Without the IDS-related concerns over accuracy and false positives, we can thus rely on them to issue preventative responses with confidence. Unfortunately, however, intrusion prevention systems do not have a silver bullet for this problem; they may in fact use the same detection methods as IDS.

The solutions provided by IPS products therefore attempt to sidestep the problem of false positives by only blocking those attacks that can be detected with high certainty. In effect, this means transfer of the strongest and most reliable technologies from the IDS domain into a different mode of operation. Even then, IPS cannot be regarded as a fix of the problem of false alarms. The solutions will seldom work perfectly "out of the box"; most require tuning to tailor their most effective operation.[7]

The biggest advantage of intrusion prevention is its potential to respond in real time and to nip attacks in the bud. However, as promising as it sounds, there are concerns about the IPS approach. The first is the overhead they can introduce in networks and systems by having to authorize all traffic and all system calls. This becomes more significant in busy

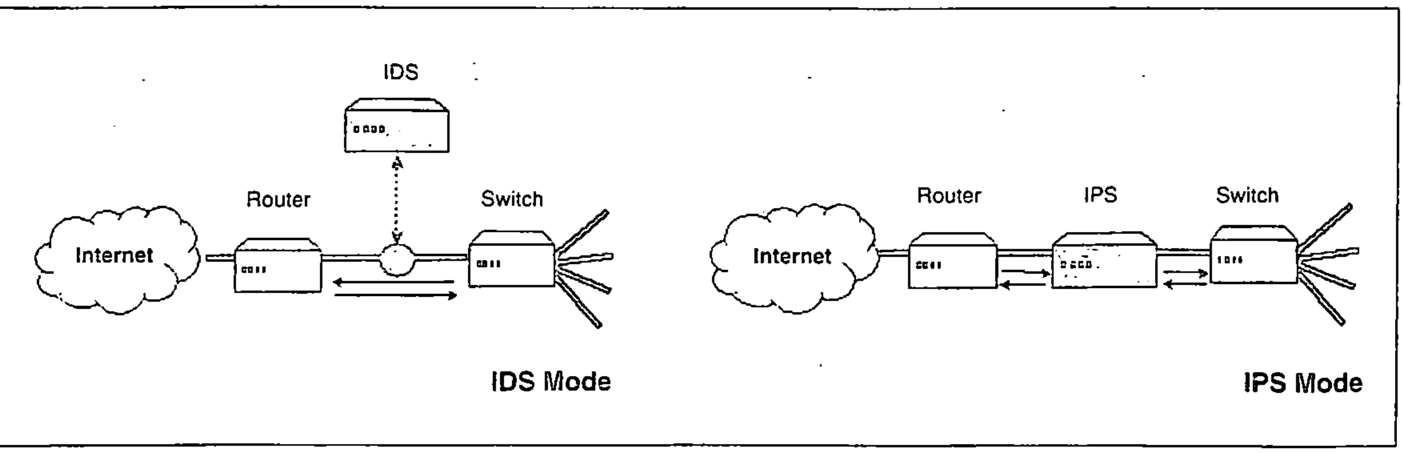


Figure 1: Offline vs. inline placement

networks and servers, where performance is crucial.[8] At the same time, however, there are parallels with firewall technologies, and devices can be designed and deployed with performance considerations in mind.

Single point of failure

A potentially worse problem is that IPSs are a single point of failure. An error here could have significant impact upon systems and networks. For example, if an IPS crashed because it couldn't cope with the traffic, or was the target of an attack, the disturbance on the network's operation would be considerable. There are some moves to overcome this problem (e.g. using a back-up IPS that takes over n an emergency, or reconfiguring the outer to redirect traffic around the IPS, or pre-configuring the IPS to run with ninimum capabilities, allowing all traffic o pass), these solutions do not fully address the issue because systems may be unprotected.

The problem of volume-related crashes s well known. Vendors are improving heir products, but still have issues to ddress. So far, a good solution remains lusive.[9]

A more significant concern is to avoid alse positives. Killing only the most suspect attacks means that a range of different attacks may pass because the cautious PS does not recognize them as intrusions. In this scenario, a further line of efence is still very desirable.

So, to answer the question posed at the ead of this section, IPS is not an alternave to IDS; it is not meant to be. But the echnologly does provide another layer of ecurity, which is important in a defence in epth strategy. As such, both IDS and IPS ave important roles, and it should not be ne case that one is used in place of the ther.

mproving our response

ut it is essential that either approach rovide a correct response to the intruon. Indeed, the fact of IPS and its tractiveness is closely linked to the need respond.

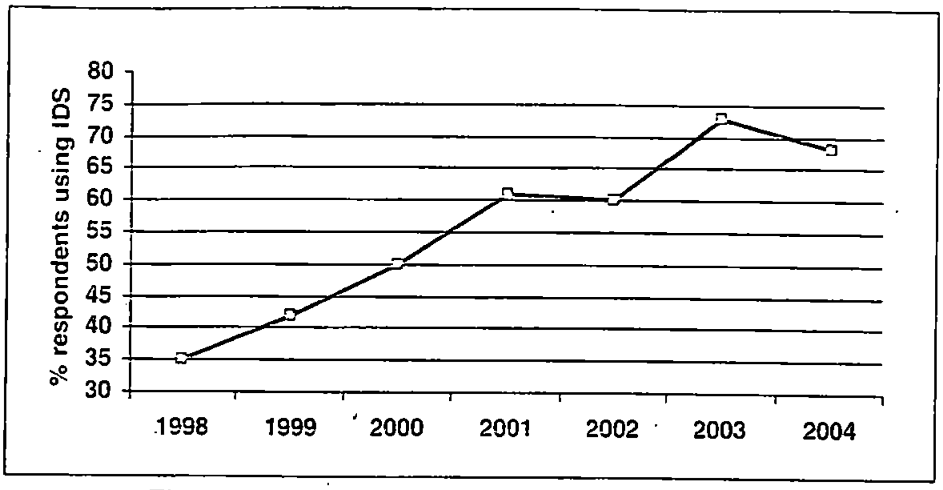


Figure 2 : Organisations using IDS technology (source: CSI/FBI surveys)

However, blocking sessions and dropping packets is not the only appropriate response. It should be possible to limit more subtle attacks by investigating them further to reach a more informed judgement.

Indeed, an IPS configured with a simple "block or pass" strategy may not provide enough flexibility; the fact that they respond only to the most definite signatures means that false negatives can occur. As such, it would be unwise to consider IPS as the only defence against intrusions. It therefore makes sense to subject the traffic that gets through to further analysis using an IDS.

At this point, however, the question still remains about what the IDS should actually do if it finds something it believes to be intrusive. If the technology is simply used in its current form, then the need to limit responses to passive actions for fear of false positives will be largely unchanged.

An alternative strategy is to endow the IDS with a sense of its own inadequacy. In other words, account for the fact that some detection judgements are likely to be stronger than others, and allow flexible levels of response to be issued accordingly.

Our group and others have researched this. Given that today's commercial IDS technologies are rooted in research from the 1980s, we should consider how current research may help to advance the commercial incarnations in the future. Our results suggests that incorporating two aspects is particularly desirable:

- Adaptation of responses according to the current context.
- Assessment of the appropriateness of response actions before and after initiating them.

Adaptive decision-making relates to the need for response decisions to change with the context in which an incident has occurred (i.e. a response that is appropriate to a one type of incident on one occasion will not necessarily be appropriate if the same incident happened again under different circumstances).

When considered in terms of two collaborating IDS entities, a Detection Engine and a Responder, the determination of this context may include a number of considerations. These include

- Whether the incident is part of an ongoing series of attacks (e.g. how many targets have already been affected? Which responses have already been issued?) .
- The current status of the target (e.g. is it a business critical system? What is its load at the moment? Is there any information or service that needs to be protected? What software/hardware can be used for response?).
- The perpetrator of the attack (is there enough information to suggest a specific attacker? Is he/she an insider/outsider?).
- The privileges of the user account involved (e.g. what is the risk of damage to the system?).
- The probability of a false alarm (how reliable has the sensor/source that

detected the incident been in the past? What is the level of confidence indicated by the Detection Engine about the occurrence of an intrusion?)

the probability of a wrong decision (how effective has the Responder been so far? Have these responses been applied before in similar circumstances?).

Having assessed the above factors, sponse decisions must then be adapted the context accordingly. For example, the incident has been detected on a isiness critical system, but the etection Engine has indicated a low nfidence, then the selection of a sponse with minimal impact upon the stem would represent the most sensible urse of action (i.e. minimizing the ance of critical operations being dispted in the case of a false positive). owever, if the same scenario occurred in njunction with previous alerts (i.e. owing that the current incident is part a series of attacks), then a more severe ponse is warranted.

The other required feature is the ability assess the appropriateness of response ions. There are two ways to do this, stly by considering the potential side ects of a response action before issuing and secondly by retrospectively alyzing its effectiveness in containing combating attacks.

The problem of side effects is a particr concern when using active responses g. blocking, termination and access trictions) because they may disrupt itimate users. As a result, the response eds to be considered before a given ion is executed. Several characteristics uld be relevant to consider in this itext:

The transparency of the response action. In some cases it might be preferable to issue responses that do not alert the attacker to the fact that he/she has been noticed; in others it could be preferable to respond very explicitly.

The degree to which the action would disrupt the user against whom it is issued. This is especially relevant when a response is mistakenly issued against a legitimate user. In situations where the Detection Engine has flagged an incident, but expressed low confidence, it may be desirable to start by issuing responses that a legitimate user would be able to overcome easily.

- The degree to which the action would disrupt other users, or the operation of the system in general. Certain responses (e.g. termination of a process, restriction of network connectivity) would affect more than just the perceived attacker. As such, the Response Policy may wish to reserve such responses only for the most extreme conditions.

Each of these factors needs to be assessed independently, and incorporated into the response selection process as appropriate, as well as during the formulation of the Response Policy by the system administrator.

The second factor that would influence appropriateness is whether a suspected attack has been used before in the same context. If the Responder keeps track of its previous response decisions, then they can be used later to assess whether the selected actions were actually effective. This requires a feedback mechanism that can then be used to refine the Response Policy.

Feedback could be provided in two ways: explicitly by a system administrator, and implicitly by the Responder itself. In the former, the administrator would inspect the alert history and manually provide feedback in relation to the responses that had been selected. This would say whether or not they had been effective or appropriate to the incident.

Otherwise, the Responder itself could infer whether previous responses had been effective. For instance, it could say whether it had been required to issue repeated responses in relation to the same detected incident. If this was true, it could potentially infer that (a) the initial response actions were not effective against that type of incident, and (b) the last response action issued might form a better starting point on future occasions (i.e. upgrading and downgrading the perceived effectiveness of the responses when used in that context).

We have a prototype implementation of the above approach. It forms part of PhD research work within our group.[12] It has provided a practical proof of concept for the ideas expressed here, and suggests that the long-term choice in the intrusion-handling domain may be broader than current detection and prevention technologies.

Having said this, we need to do some more work on it before it is ready for large-scale deployment.

Conclusion

The title of this article was perhaps a little misleading, in the sense that neither technology is a complete answer. Although IPS technologies provide a way to thwart high-certainty attacks, we still need the IDS to account for other cases.

This leaves us with a problem. The imperfect nature of detection means that we can easily mistake normal activity for an intrusion. At the same time manual responses could be too late to prevent incidents.

We advocate a more flexible and intelligent approach, one that offers escalating levels of response according to several contextual factors. Although we have worked on this with some effect, we need to do more to reduce the uncertainty in response decisions before we look to automate fully prevention and response activities.

About the authors

Maria Papadaki has recently completed her PhD research within the Network Research Group, focusing upon the issue of flexible, automated IDS response. This research activity was undertaken with support from the State Scholarship Foundation of Greece.

Dr Steven Furnell is head of the Network Research Group at the University of Plymouth, UK. He is author of "Cybercrime: Vandalizing the Information Society", published by Addison Wesley.

ferences:

Cohen F.B. 1999. "Simulating Cyber
acks, Defences, and Consequences",
e Infosec Technical Baseline studies,
rch 1999. http://all.net/journal/ntb/
ulate/simulate.html.

etwork Associates. 2003. The Path to
vention, White Paper, Network
ociates Technology, Inc, October
3. http://www.nai.com/

artner. 2003. "Gartner Information
urity Hype Cycle Declares Intrusion
ection Systems a Market Failure",
tner Press Release, 11 June 2003.

ylor S. and Wexler J. (2003) "IDS vs.
Is one strategy 'better?'", Network
ld Fusion, 16 October 2003.
://www.nwfusion.com/newsletters/
e/2003/1013wan2.html

penService, Inc. 2003 Security Event
agement Survey Results Analysis:
ght into the Threats, Issues and Trends Facing Network Security Departments in 2003. White Paper. February 2003. www.open.com.

[6] Gordon, L.A., Loeb, M.P., Lucyshyn, W. and Richardson, R. 2004. Ninth Annual CSI/FBI Computer Crime and Security Survey. Computer Security Institute.

[7] McAffee Security. 2003. Intrusion Prevention: Myths, Challenges, and Requirements. White Paper. Network Associates. April 2003.

[8] Messmer, E. 2002. "Intrusion prevention' raises hopes, concerns", Network World Fusion, 4 November 2002, http://www.nwfusion.com/news/2002/1104prevention.html

[9] Snyder, J. 2003. "False positives remain a major problem", Network World Fusion Online Magazine, 13 October 2003.http://www.nwfusion.com/reviews/2003/1013idsalert.html

[10] Papadaki M., Furnell S.M., Lines B.M., and Reynolds P.L. 2003. "Operational Characteristics of an Automated Intrusion Response System", in Communications and Multimedia Security: Advanced Techniques for Network and Data Protection Lioy A. and Mazzochi D. (eds), Springer Verlang, October 2003: pp 65-75.

[11] Ragsdale, J.D., Carver, C.A. Jr., Humphries, J.W., and Pooch, U.W. 2001. "Adaptation Techniques for Intrusion Detection and Intrusion Response Systems", 2nd Annual IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop, West Point New York, June 5-6 2001.

[12] Papadaki, M. 2004. Classifying and Responding to Network Intrusions. PhD Thesis. University of Plymouth, United Kingdom.

Automating the process of intrusion response

M.Papadaki and S.M.Furnell
Network Research Group, School of Computing, Communications & Electronics, University of Plymouth,
Plymouth, United Kingdom
nrg@plymouth.ac.uk

Abstract

The volume and speed of Internet-based attacks means that automated intrusion response is becoming essential. However, effective automation is complicated by the potential for issuing severe actions in a false positive scenario. Addressing this problem leads to requirements such as the ability to adapt decisions according to changes in the environment, the facility to offer escalating levels of response, and the capability to evaluate response decisions. The paper discusses how these concepts are achieved within the Flexible Automated Intelligent Responder (FAIR) architecture, and then outlines how response policies may be used to enable the desired degree of autonomous action.

Keywords

Intrusion Detection, Intrusion Response, Automated Response, False Alarm

INTRODUCTION

Despite the increasing adoption of security technologies, the problem of intrusions continues to plague networked systems. The volume of reported incidents is rising (CERT 2003), and the associated financial losses remain significant (Gordon et al. 2004). Although Intrusion Detection Systems (IDS) are available to highlight problems, the issue of response must also be considered, and capabilities here have traditionally been limited to alerting the administrator about the occurrence of a suspected incident. However, responding manually to intrusions can be problematic, especially in the case of large networks, with a high number of IDS notifications and alerts. It may also not be practically feasible, given the speed and escalation of many attacks. As a result, there is an increasing trend to automate response and enable systems !o issue automatically more severe actions such as, terminating connections, or blocking attacking sources.

Unfortunately, automating response is not a straightforward solution, as issuance of the aforementioned active responses in a false alarm scenario could disrupt legitimate users, and affect their level of access. Hence it is important to ensure that responses launched without prior human authorisation have the predicted effects and do not put the system at greater risk.

This paper considers the desirable characteristics of an automated response system, and outlines a novel architecture for fulfilling them. The process of flexibly selecting response actions is considered, leasing to discussion of how associated response polices are realized within a prototype implementation.

OPERATIONAL REQUIREMENTS FOR AN AUTOMATED RESPONDER

Current IDS products approach response by enabling the prevention of threats that are not prone to false alarms. Additionally, research efforts have made valuable contributions in terms of how the cost of an attack can be balanced against the cost of response (Balepin et al. 2003; Toth and Kruegel 2002), and how the probability of a false alarm can influence response decisions (Carver et al. 2001). However, there is still scope for improvement. In order for an automated response system to be further enhanced, it should provide greater flexibility for the response mechanism to not only prevent threats, but include actions that aim to investigate, minimise the threat at the target, forestall the occurrence of future threats, etc. The following operational characteristics of a response system are considered important in order to provide a foundation for such requirements.

Adapt decisions to account for changes in the environment

An important characteristic is the ability to vary decisions depending upon the context in which the incident has occurred. Thus it is important to account for changes in the environment and adapt response decisions accordingly. Specifically, the Responder can draw upon information from a number of sources in a network, in order to assess the overall context in which an incident has occurred, including considerations such as:

- the overall alert status of the IDS at the time of the new incident;

- whether the incident is part of an ongoing series (e.g. how many targets have already been affected? Which responses were already issued?);
- the current status of the target (e.g. is it business critical? What is its current load? Are any active users connected? Is any software running that introduces additional risk? What software/hardware can be used for response?);
- the privileges of the user account involved (e.g. what is the risk of damage to the system?);
- the probability of a false alarm (how historically reliable is the sensor/source that detected the incident? What is the intrusion confidence level from the Detection Engine?);

Having assessed the above factors, response decisions must then be adapted to the context accordingly.

Offer flexible and escalating levels of response

Another desirable feature is the ability to offer escalating levels of response to account for the varying levels of threat. The main advantage of this approach is that it overcomes the problem of existing IDS solutions, which can either permit or deny a security event, and do not offer any scalability, which is essential for cases of false positive alarms. The proposed system should support a wider variety of response actions (other than stop the attack and restore the system), enabling a greater level of flexibility for a variety of events. For example, in the event of a suspicious user login, the Responder can select responses of varying severity, according to the circumstances. Possible options would include just logging the event and doing nothing else, alerting the administrator, allowing the user to login but increasing the monitoring of activity, allowing the login but limiting access rights to prevent potential damage to the system, issuing an explicit authentication request before allowing the login, or denying access altogether.

The basis for achieving scalable responses is the assessment of the context of an attack (including the overall threat introduced by the occurrence of the attack), and the estimation of the impact of a response action.

Evaluate response decisions

The appropriateness of responses should be evaluated before and after initiation. Specifically, the main considerations in evaluating a response are based upon its potential side effects, and its practical effectiveness in fulfilling its intended role.

Side effects are a particular concern in the context of active responses, because they have the potential to adversely affect legitimate users. As such, this needs to be considered before the Responder initiates a given action. There are a number of characteristics that would be relevant in this context, including the transparency of the response action, and the degree to which it would cause disruption if mistakenly issued against legitimate user(s).

The practical effectiveness of the response can be reflected in its efficiency. This requires some form of feedback, which could be provided in two ways: explicitly by a system administrator, and implicitly by the Responder. In the former case, the administrator would inspect the alert history and manually provide feedback regarding responses that had been selected, to indicate whether or not they had been effective or appropriate to the incident. In case of badly issued responses, the administrator could indicate whether the response: was too severe; was not severe enough; had unwanted side effects; was applied too late; or was completely inappropriate. This information would be used to adjust the effectiveness of the response action and the decision capability of the Responder.

By contrast, the implicit feedback would require the Responder to infer whether previous responses were effective. A simplified example of this would be to determine whether it had been required to issue repeated responses for the same detected incident. If this was the case, then it could potentially infer that the initial response actions were not effective, and the last action issued might form a better starting point on future occasions.

Having obtained such feedback, it would be desirable for the system to automatically incorporate it into a refined response policy. This, however, would be a non-trivial undertaking, and it is anticipated that machine-learning mechanisms would be required to facilitate a fully automated process (Mitchell 1997). An alternative would be to collate the feedback, and present it to the system administrator for later consideration when performing a manual overhaul of the Response Policy.

THE FAIR ARCHITECTURE

Having identified these requirements, it is necessary to consider a response framework within which they can be achieved. As such, the conceptual framework for a Flexible Automated Intelligent Responder (FAIR) is proposed. FAIR is based upon the concept of a centralised host handling the monitoring of a number of networked client systems. The core elements are illustrated in Figure 1, and discussed below.

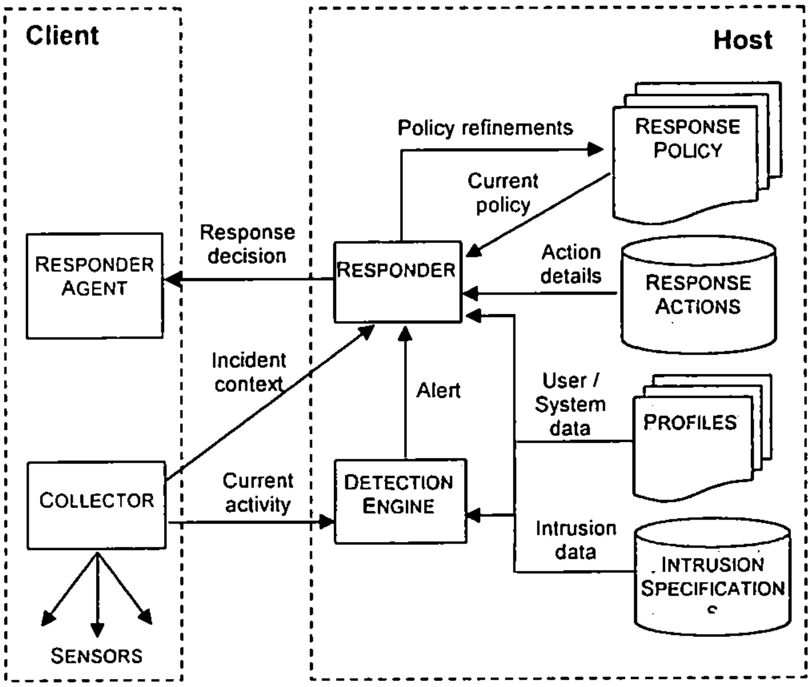


Figure 1 : The FAIR architecture

- **Detection Engine**: Analyses current activity and raises alerts for suspected intrusions. Informs the *Responder* of the intrusion type, along with factors such as the target of the attack, and perceived perpetrator.
- **Responder**: Monitors alerts, considering them in conjunction with incident context to take appropriate actions where necessary.
- **Collector**: Provides initial activity data to *Detection Engine*, and subsequently informs *Responder* about current context on the target system (e.g. applications running, active network connections, processor load).
- **Intrusion Specifications**: *Intrusion Specifications* contain information about specific types of intrusions and their characteristics, such as incident severity rating, ratings of likely impacts (e.g. in terms of confidentiality, integrity and availability), and the speed with which the attack is likely to evolve.
- **Profiles**: Contain data about users, systems, and attackers, which can provide additional context for response decisions.
- **Response Actions**: Details of available response actions, enabling selection of responses with the most appropriate characteristics (e.g. stopping power, transparency).
- **Response Policy**: Uses expert systems technology to indicate the most desirable characteristics for responses in the current context.
- **Responder Agent**: Initiates and manages any response actions required on the target (e.g. correcting vulnerabilities, authentication challenges, limiting access rights).

As seen from the figure, the Responder uses information from several sources to assist its decisions. This information is used to determine the context of an attack, and examples of the measured factors include Overall Threat, Alert Status, Alarm Confidence, Target Importance, and Responder Efficiency (Papadaki 2004). The next section considers how these are used to inform the selection of response actions.

SELECTING RESPONSE ACTIONS

In order to enable the selection of appropriate responses, information is required for each available action, to enable its characteristics to be matched to the current context.

Phase

The Response Phase reflects the main objective of a response, which at a high level may be:

- Notify
- Investigate
- Protect Resources
- Recover
- Collect Evidence, and
- Forestall potential problems

Each response action is associated with one of the above phases. The Responder will decide which phase of response is more appropriate for the specific alert, by combining information from contextual factors. For example, if the probability of a false alarm was not low and the Urgency to respond was not high, then the best Phases of response would be to Investigate, and Collect Evidence (i.e. passive responses). If the probability of a false alarm was low and the decision capability of the Responder was high, then the best Phases would be Protect, Collect Evidence, and Recover (i.e. increased likelihood of active responses being involved).

Stopping Power

This metric reflects the perceived strength of a response against the attack. It is represented on a 10-point scale, where the different ratings are derived according to Table 1.

Response Stopping Power	
9-10	Block / Stop Attack
7-8	Redirect
5-6	Stop partially / Limit
4	Postpone, Delay
3	Investigate
2	Collect Evidence
1	Minimise loss at target
0	None

Table 1 : Response Stopping Power Ratings

In another implementation scenario, where the same response could be used for more than one purpose (e.g. Delay attack, and Investigate), separate ratings for each purpose could be applied. For example, a response could have a rating of 7 for partially stopping the attack, and 5 for delaying it. However, for simplicity, each response has only one rating, which reflects the main purpose of the response.

The Responder selects the maximum Stopping Power allowed. This is determined within the Response Policy, and influenced by the context of the attack. The higher severity the context is, the higher the maximum Stopping Power can be.

Counter-Effects

The Counter-Effects is one aspect of the side effects of a response action, in terms of the impact they can have upon the confidentiality, integrity, and availability of systems and data. Responses that might give in information about the presence of the IDS, software versions, system vulnerabilities, or important assets of a system will have impact upon the confidentiality of that system. Responses that affect the integrity of systems, especially in the case of deceptive responses (where false information is given to suspected attackers), or recovery responses, usually have high integrity counter-effects. Finally, responses that deny access to systems, processes, or data, normally have high availability counter-effects.

As the Responder selects the most appropriate characteristics of candidate responses, the counter-effects should be as low as possible. So, if two response actions with similar characteristics are selected, the one with lower side effects will receive higher level of confidence. Considering the counter-effects of a response is particularly important in case of false alarm scenarios. Generally, low counter-effects are considered important when the

Overall Threat introduced by the incident and the Urgency to Respond are low, as there is no pressing need for the FAIR system to interfere significantly. By contrast, when these metrics are high, the Responder will need to select more effective responses, even ones with high counter-effects.

Transparency

This is the other aspect relating to response side-effects, and reflects how apparent a response will be to the attacker, legitimate users, or the system overall. Responses such as collecting evidence have high transparency, as users are not aware of them being issued. By contrast, explicit authentication requests have low transparency, as the user cannot proceed with activities unless he replies. The aim of the Responder is to select responses with as high transparency as possible. Like the counter-effects, transparency is particularly important when the Urgency to respond, the Overall Threat introduced by the incident, and the probability of a true alarm are not high.

Response Efficiency

The Response Efficiency reflects the overall effectiveness of the response action, based upon its historical performance, and specifically the feedback received whenever it is issued. It is thus refined gradually, to reflect additional feedback from new alerts. The higher the Response Efficiency is, the more suitable a candidate response can be. As already discussed, the efficiency of responses becomes an increasingly important consideration, in high severity scenarios, where the Urgency to respond and the Overall Threat introduced by the incident are high.

RESPONSE POLICY

Within the FAIR architecture, the ways in which the aforementioned factors can influence a response decision is expressed within a response policy. This is essentially an expert system module, responsible for identifying the most appropriate response characteristics, according to the context of the attack. Following an alert, the Response Policy receives the static and dynamic context of the attack, and it selects which characteristics would be more suitable for the selected responses.

As a starting point, the first characteristic identified is the Response Name. The administrator is able to fully customise this part of the selection process, via a user-friendly interface that allows the association of responses with attacks, and possible conditions under which they can be selected. For example, the Response Policy could define rules, shown in Figure 2 (note: although shown in the example, the presence of conditions is not necessary, leaving the flexibility for the use of as simple or as complicated response policies as possible).

```
"If the Alert relates to a Buffer Overflow attack,
 and the Target is Vulnerable, then
     Deny / Stop the action,
     Patch the Vulnerability immediately"
"If the Alert relates to a Buffer Overflow attack,
 and the Target is Not Vulnerable, then
     Check how many systems are vulnerable,
     Patch the Vulnerable Systems as soon as possible"
```

Figure 2 : Response Policy rules

Based upon the user-defined policy, a number of responses are selected as candidates. Then, more general rules are applied to determine the remaining response characteristics. These rules are not specific to the different types of attacks, and cannot be as easily customised by the administrator (and indeed should not need to be). Overall, there are three sets of rules, which aim to determine:

- the most appropriate Response Phases,
- the maximum level of Stopping Power allowed, and
- how important the Response Efficiency should be, in comparison to the response side effects (Transparency and Counter-Effects).

The rules to select the most suitable Response Phases are mainly influenced by the response capability of the Responder (Responder Efficiency), the Overall Threat introduced by the incident (Overall Threat), Urgency to respond (Urgency), and probability of a true alarm (Alarm Confidence), as illustrated in Figure 3.

```
“If Responder Efficiency is Low,
 and Overall Threat is High, then
      suitable phase is Notify”
“If Alarm Confidence is Low,
 and Urgency is Low, then
      suitable phases are Investigate and Collect Evidence”
“If Alarm Confidence is High,
 and Responder Efficiency is High, then
      suitable phases are Protect Resources, Collect Evidence,
      and Recover”
```

Figure 3 : Examples rules for selecting response phases

The first rule suggests that notification alerts are suitable for cases when severe responses are needed, but the Responder is not able to issue them automatically (because the Responder Efficiency is low). The notified administrator will review the decision and authorise any severe responses that the Responder could not issue. The second rule suggests that if there is a suspected attack with low confidence, and the probability of it escalating rapidly is low, then the best phase of response would be to Investigate and Collect Evidence. Thus, more information will be collected, to determine whether the attack is really occurring. Finally, the third rule suggests that, if the Alarm Confidence and the Responder Efficiency are both High, then the most suitable phases are to Protect Resources, Collect Evidence, and Recover.

The maximum level of Stopping Power is influenced by the Responder Efficiency, the general Alert Status of the system, Urgency, Overall Threat, and Target Importance. Effectively, the higher these factors are, the higher the Stopping Power can be.

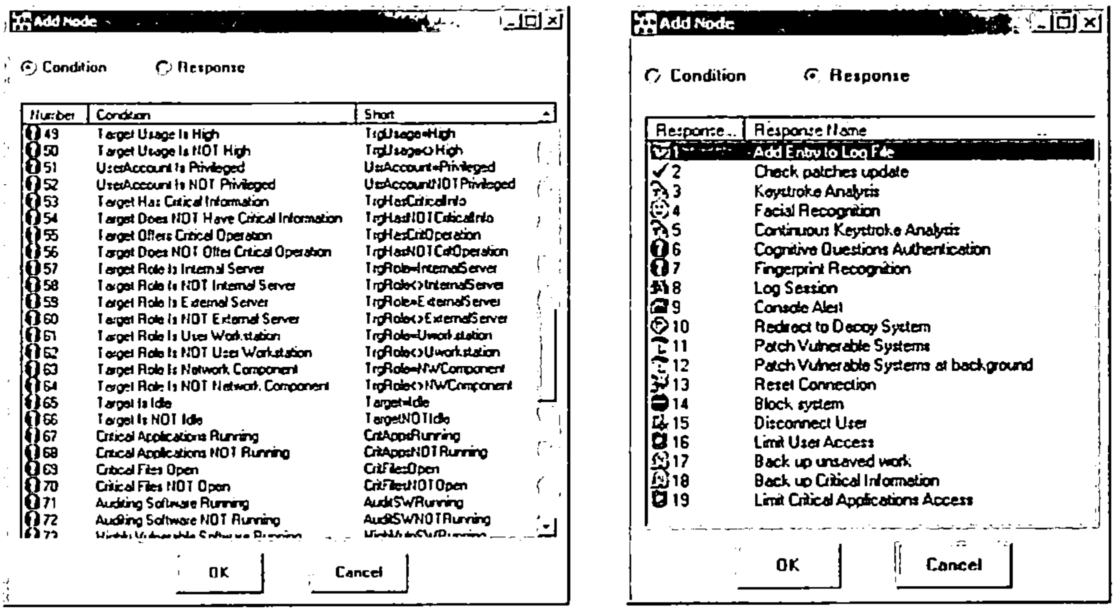
Finally, the last part of the selection process is to determine the weighting of the Efficiency and side effects of a response. Influencing factors for that process are the Overall Threat and Urgency. The higher these are, the more important it is for responses to be efficient, and the lower they are, the more important it is for responses to have low side effects.

After the ‘ideal’ response characteristics are determined, the candidate responses (the responses already selected in the first phase) are given confidence metrics, according to how closely they match the desired response characteristics. The strongest choices (the ones with confidence higher than 50%) are included in the list of approved responses, whereas the rest remain in the category of candidate responses. The approved responses are the ones to be issued automatically by the Responder, while the candidate responses are the ones that are included in the alert as a reference for the administrator, effectively awaiting human authorisation before they can be issued.

PROTOTYPE REALISATION

The aforementioned process has been realised in a proof-of-concept prototype, which demonstrates the main characteristics of the FAIR architecture. As part of this implementation, a Response Policy Manager (RPM) tool provides a user-friendly interface for the review of policy rules, which are represented via a hierarchical tree, where the types of alerts are at the highest level and the response actions lie at the lowest levels. The RPM allows the use of intermediate branches in the tree, which comprise the conditions under which specific response actions are initiated for particular alerts. When the administrator selects a specific policy that the Responder should use to make decisions, the RPM generates the rules corresponding to the policy's tree structure, and updates the expert system's rules.

Creating, or modifying a policy involves the addition, edit, and removal of nodes from the tree structure. Having already loaded the available alerts (this is done automatically by creating a new policy), each node added or edited can either be a ‘Condition’ (Figure 4a), or a ‘Response’ (Figure 4b).



Add Node

● Condition ○ Response

Number	Condition	Short
49	Target Usage Is High	TrgUsage=High
50	Target Usage Is NOT High	TrgUsage<>High
51	UserAccount Is Privileged	UsrAccount=Privileged
52	UserAccount Is NOT Privileged	UsrAccountNOTPrivileged
53	Target Has Critical Information	TrgHasCriticalInfo
54	Target Does NOT Have Critical Information	TrgHasNOTCriticalInfo
55	Target Offers Critical Operation	TrgHasCritOperation
56	Target Does NOT Offer Critical Operation	TrgHasNOTCritOperation
57	Target Role Is Internal Server	TrgRole=InternalServer
58	Target Role Is NOT Internal Server	TrgRole<>InternalServer
59	Target Role Is External Server	TrgRole=ExternalServer
60	Target Role Is NOT External Server	TrgRole<>ExternalServer
61	Target Role Is User Workstation	TrgRole=Uworkstation
62	Target Role Is NOT User Workstation	TrgRole<>Uworkstation
63	Target Role Is Network Component	TrgRole=NWComponent
64	Target Role Is NOT Network Component	TrgRole<>NWComponent
65	Target Is Idle	Target=Idle
66	Target Is NOT Idle	TargetNOTIdle
67	Critical Applications Running	CritAppsRunning
68	Critical Applications NOT Running	CritAppsNOTRunning
69	Critical Files Open	CritFilesOpen
70	Critical Files NOT Open	CritFilesNOTOpen
71	Auditing Software Running	AuditSWRunning
72	Auditing Software NOT Running	AuditSWNOTRunning
73	Highly Vulnerable Software Running	HighVulnSWRunning

(a)

Add Node

○ Condition ● Response

Response...	Response Name
1	Add Entry to Log File
2	Check patches update
3	Keystroke Analysis
4	Facial Recognition
5	Continuous Keystroke Analysis
6	Cognitive Questions Authentication
7	Fingerprint Recognition
8	Log Session
9	Console Alert
10	Redirect to Decoy System
11	Patch Vulnerable Systems
12	Patch Vulnerable Systems at background
13	Reset Connection
14	Block system
15	Disconnect User
16	Limit User Access
17	Back up unsaved work
18	Back up Critical Information
19	Limit Critical Applications Access

(b)

Figure 4 : Adding (a) Conditions and (b) Responses

The culmination of this process is illustrated in Figure 5, which shows an example of the response rules that could be specified for an 'Authentication Failure' alert. In this case, the Responder would consider checking for the most recent update of related software to ensure that it is not vulnerable, initiating keystroke analysis and facial recognition (if available) to authenticate the user in a non-intrusive manner. The conditions for the latter to happen would not be just the occurrence of the incident. Only the addition of the alarm to a log file would happen in that case. For the previously mentioned responses to be issued, then the policy rules in Figure 5 require that the alarm confidence is low (hence the Responder would need to collect more information about the incident), and the overall threat and the importance of the target should be low as well, not justifying the issue of more severe responses. Also, the account involved would need to be not privileged, with login time outside the normal pattern, in order to issue non-intrusive authentication. If a privileged account is involved, then the policy shows a more elaborate range of responses being suggested.

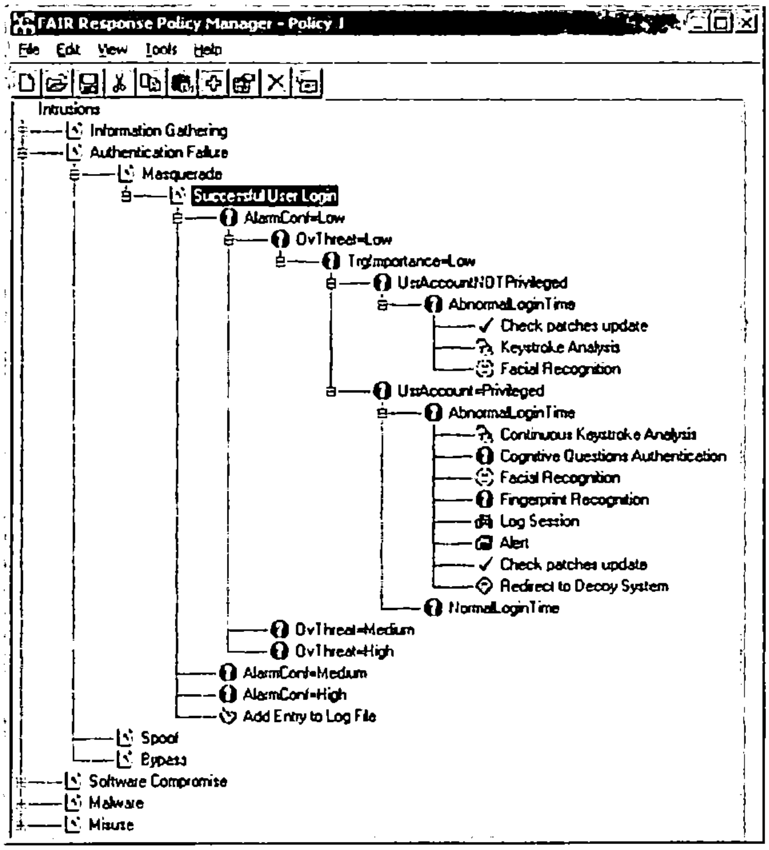


Figure 5 : FAIR Response Policy Manager

CONCLUSION

This paper has outlined the two main requirements of a flexible intrusion response architecture: the ability to adapt decisions according to changes in the environment, and the ability to provide easily customisable response policies. The latter was evidenced through an overview of the Response Policy Manager.

The wider prototype has served to prove the viability of flexible automated and intelligent response. The system in its current form is believed to represent an advancement on existing intrusion response approaches, and the decision making capability of the prototype has been tested against intrusion scenarios created using an attack simulation console. The logical next stage of the research will be to extend the work and evaluate effectiveness against real intrusion scenarios.

REFERENCES

Balepin I., Maltsev S, Rowe J, and Levitt K. (2003) "Using Specification-Based Intrusion Detection for Automated Response", in *Proceedings of the 6th International Symposium RAID 2003 (Recent Advances in Intrusion Detection)*, Pittsburgh, PA, September 8-10, 2003.

Carver C.A. Jr., Hill J.M.D, and. Pooch U.W. (2001) "Limiting Uncertainty in Intrusion Response", *2nd Annual IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop*, West Point New York, June 5-6 2001.

CERT/CC. (2003) "CERT/CC Statistics 1988-2003", CERT Coordination Center, URL http://www.cert.org/stats/cert_stats.html, Accessed 31 July 2004.

Gordon, L.A., Loeb, M.P., Lucyshyn, W. and Richardson, R. 2004. *Ninth Annual CSI/FBI Computer Crime and Security Survey*. Computer Security Institute.

Mitchell T.M. (1997) *Machine Learning*, McGraw-Hill, New York, 1997.

Papadaki, M. (2004) Classifying and Responding to Network Intrusions, PhD Thesis, University of Plymouth, Plymouth, United Kingdom.

Toth T. and Kruegel C. (2002) "Evaluating the impact of automated intrusion response mechanisms", in *Proceedings of the 18th Annual Computer Security Applications Conference (ACSAC)*, 9-13 December 2002, San Diego California, IEEE Computer Society Press, USA.

COPYRIGHT

[M.Papadaki, S.M.Furnell] ©2004. The author/s assign the We-B Centre & Edith Cowan University a non-exclusive license to use this document for personal use provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive license to the We-B Centre & ECU to publish this document in full in the Conference Proceedings. Such documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. Any other usage is prohibited without the express permission of the authors

Advanced Authentication and Intrusion Detection Technologies

Paul Dowland, Dr Steven Furnell, George Magklaras, Maria Papadaki, Prof Paul Reynolds, Philip Rodwell, Harjit Singh
Network Research Group, Department of Communication and Electronic Engineering, University of Plymouth, UK

Abstract

Security is a vital consideration in the age of modern networks and Internet-based communications, and represents an essential underpinning of emerging applications such as electronic commerce. Within this domain, the ability to ensure the authorised and correct use of systems is an area of significant challenge. The research to be presented is centred around the Intrusion Monitoring System, a conceptual architecture for real-time user authentication and supervision, which has been defined by an earlier project within the Network Research Group. The current research encompasses advanced authentication technologies, based upon biometric techniques and user behaviour profiling. These approaches improve considerably upon the traditional user name and password combination, which has been proven to be weak and susceptible to compromise. While enhanced authentication will combat external impostors and internal masqueraders, further research addresses methods of identifying system misuse originating from authorised users, whom independent surveys have established account for around 80% of computer abuse incidents. Another important consideration relates to methods of responding to suspected intrusions in a manner that will trap genuine impostors and misfeasors, without unduly disrupting legitimate user activity in cases where a false classification has occurred. The research considers the application of these authentication and intrusion detection approaches within both traditional desktop environments and third generation mobile networks.

The Intrusion Monitoring System architecture

The Intrusion Monitoring System (IMS) is the focus of security research in the Network Research Group. IMS is an architecture for intrusion monitoring and activity supervision, based around the concept of a centralised host handling the monitoring of a number of networked client systems. Intrusion detection is the system is based upon the comparison of current user activity against both historical profiles of 'normal' behaviour for legitimate users and intrusion specifications of recognised attack patterns. The architecture is comprised of a number of functional modules, addressing data collection and response on the client side and data analysis and recording at the host (as illustrated in the figure below).

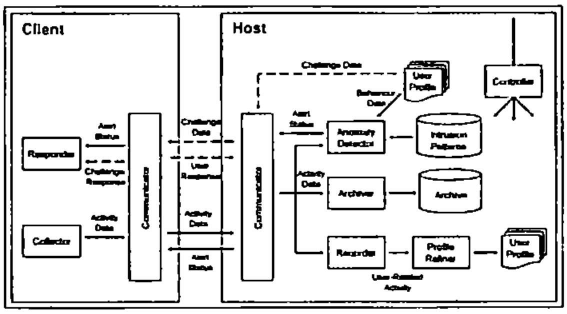


Figure 1: Intrusion Monitoring System architecture

Related Research Projects

The current IMS related research projects are listed below, a number of these involve collaboration and/or sponsorship from Orange Personal Communications Services.

1. User authentication and supervision in networked systems

This project concerns the investigation and evaluation of composite authentication techniques. The study recognises that a variety of authentication techniques are available which, when used in isolation, have known error rates in terms of false rejection and false acceptance. The research is focused upon the specification, implementation and evaluation of a composite authentication approach, in which a range of technologies are available and can be applied intelligently by the system as appropriate to the active user and their current task.

2. Behavioural profiling and intrusion detection systems using data mining

This project seeks to develop profiles of user behaviour by applying intelligent analysis techniques to system data collected in real-time during Windows NT sessions. The profile would represent a model of the legitimate users normal behaviour and could subsequently be utilised in a real-time supervision context to ensure that the activities of the user match those expected of the claimed identity. As such, the technique offers the potential to identify impostors and misusers within the system. This project contributes to the profiling aspect of the IMS architecture.

3. Generic approaches for intrusion specification and misuse detection

This project seeks to design and develop a generic intrusion specification language, which may then be used to specify intrusion characteristics in detection systems such as IMS. This work leads in to specific consideration of how to detect misfeasor attacks – that is, misuse of the system by a legitimate user. In terms of the IMS architecture, this project will contribute to the mechanisms utilised by the Anomaly Detector module.

4. Classifying and responding to network intrusions

This project seek to determine a comprehensive taxonomy of system-detectable intrusions and misuse, leading to the consideration of how the IMS system should respond to them. The work will involve design and practical evaluation of alternative response strategies, assessing factors such as the effectiveness against the nominated class(es) of intrusion and any negative effects that the response could have upon a legitimate user in a false rejection scenario.

5. Non-intrusive security for third generation mobile systems

This project highlights the need for improved methods of user authentication in future mobile systems such as the Universal Mobile Telecommunications System (UMTS). The work is focused upon an investigation of user-to-terminal and user-to-network forms of authentication for use in future mobile networks and devices.

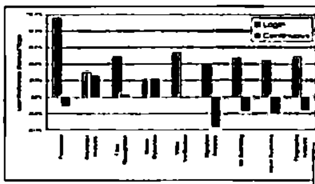


Figure 2: User preference of authentication and supervision methods (178 respondents)

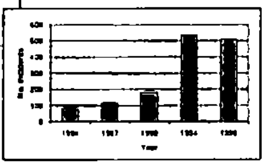


Figure 3: Reported computer crime incidents (source: *Audit Commission*)

Authentication & Intrusion Detection Approaches

IDS techniques are based on the assumption that an intruder's behaviour will be different from that of a legitimate user. In order to detect this deviation from normal user behaviour, IDSs collect audit data such as system resource usage. Providing this continuous monitoring involves processing and analysing vast amount of audit data. Hence relying on human expertise is time consuming, knowledge intensive and infeasible past a certain volume of data. Therefore intelligent data analysis techniques are required to automate some of this process. The research is investigating techniques to automate some of the data analysis using Data Mining (DM) techniques and methodologies (figure 4 illustrates initial results).

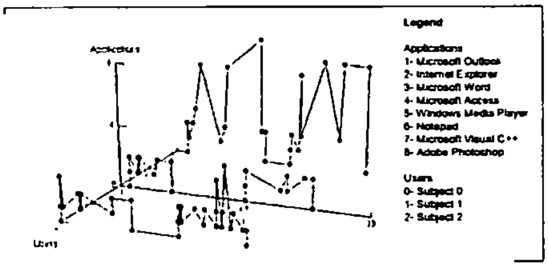


Figure 4: Graphical representation of applications run by users (IMS - Behavioural Profiling)

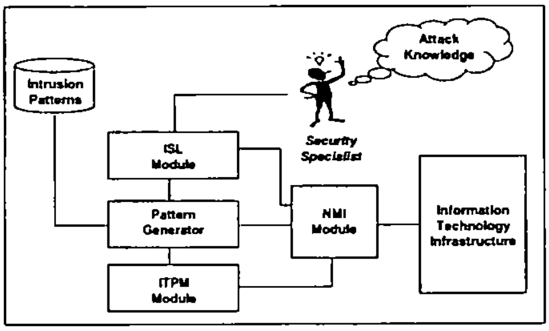


Figure 5: Functional modules of an Intrusion Specification Language

ISL module The Intrusion Specification Language module, responsible for describing intrusions in a standardized, system independent manner. Used by the security specialist/architect.

Pattern generator module It converts the ISL descriptions into system-specific patterns. It also performs optimised pattern matching functions that are essential for real-time intrusion detection.

ITPM module The Insider Threat Prediction Model tries to sense the level of sophistication of a legitimate user. It can then estimate the probability that a particular user will misuse the infrastructure. This is an experimental/new approach of tackling the insider IT misuse problem.

NMI module The Network Management Integration module is responsible for utilising network management protocols in order to collect information and co-ordinate selected IDS responses from a variety of IT infrastructure components.

Information technology infrastructure The set of computer hardware, software and telecommunication components that perform a useful function inside an organisation.

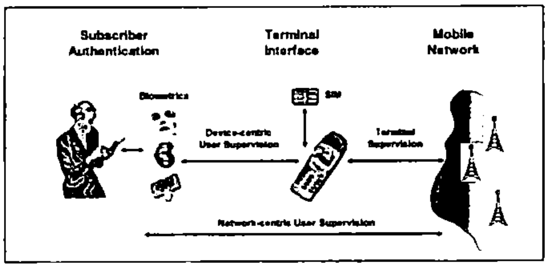

Subscriber Authentication
Terminal Interface
Mobile Network
Biometrics
SIM
Device-centric User Supervision
Terminal Supervision
Network-centric User Supervision

Figure 6: Supervision in a Mobile Environment

Summary

The techniques under investigation represent a considerable departure from traditional methods of authentication and access control, and aim to provide an added level of protection for IT systems. Modern society is increasingly dependent upon IT infrastructures. At the same time, surveys from bodies such as the Audit Commission and the FBI are reporting increased levels of computer crime and abuse (originating from both outside the organisation and from within). In view of these factors, the additional safeguards provided by advanced security techniques will become ever more necessary. As the research has identified, the techniques are relevant to both traditional networked PC environments, as well as other scenarios such as third generation mobile systems.

Dowland, P S, Furnell, S M, Illingworth, H M and Reynolds, P L 1999 "Computer Crime and Abuse: A Survey of Public Attitudes and Awareness", *Computers & Security*, vol 18, no 8, pp715-726

Dowland, P. and Furnell, S 2000 "Enhancing Operating System Authentication Techniques", *Proceedings of the Second International Network Conference (INC 2000)*, Plymouth, UK, 3-6 July 2000, pp253-261.

Furnell, S M, Illingworth, H M, Katsikas, S K, Reynolds, P L, and P W Sanders 1997, "A comprehensive authentication and supervision architecture for networked multimedia systems", *Proceedings of IFIP CMS '97*, Athens, Greece, 22-23 September 1997, pp227-238

Furnell, S M and Dowland, P S 2000 "A conceptual architecture for real-time intrusion monitoring", *Information Management & Computer Security*, vol 8, no 2, pp65-74

Furnell, S M, Dowland, P S, Illingworth, H M and P L Reynolds 2000 "Authentication and supervision: A survey of user attitudes", *Computers & Security*, vol 19, no 6, pp529-539

Papadaki, M 2000 *A Taxonomy of I.T. System Intrusions* M Sc Thesis, University of Plymouth, Plymouth, UK

Rodwell, P M, Furnell, S M and Reynolds, P L 2000 "Non-intrusive security requirements for third generation mobile systems", *Proceedings of PG Net 2000 - 1st Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting*, Liverpool, UK, 19-20 June 2000, pp7-12.

http://ted.see.plym.ac.uk/nrg

Enhancing Intrusion Response in Networked Systems

Maria Papadaki, Steven Furnell, Paul Dowland, Benn Lines, Paul Reynolds
Network Research Group, Department of Communication and Electronic Engineering, University of Plymouth, UK

Abstract

Increasing levels of attacks and misuse, as well as continuing organisational dependence upon computing and networked systems, have served to make intrusion detection systems an increasingly important security technology. However, although intrusion detection methods have received extensive research focus, the task of *responding* to attacks has received relatively little attention, particularly in the context of automated and active response systems.

The research relates to the design and practical evaluation of a novel architectural framework for intrusion response strategies, based on the identification of the range of factors influencing the response decision process and the adaptation of decisions according to custom response policies. The research is based on the Intrusion Monitoring System (IMS) architecture, and argues that a more comprehensive and reliable response framework is required in order to facilitate further automation of active responses. The poster presents a categorisation of the factors influencing response decisions, along with an overview of the proposed response architecture. It also presents details of an operational prototype system that is being developed, based upon the response architecture proposed.

Factors Influencing Intrusion Response

In order to address the automated response issue, it is important that further attention is given in the identification and assessment of factors that influence the response decision process.

Figure 1 depicts the main factors, split according to whether they relate to the incident or the IDS. The *incident* is the trigger for the response and still represents the principal influence over what should be done. However, assessment of the other factors enables the responder to establish the context of the incident, and select appropriate responses accordingly.

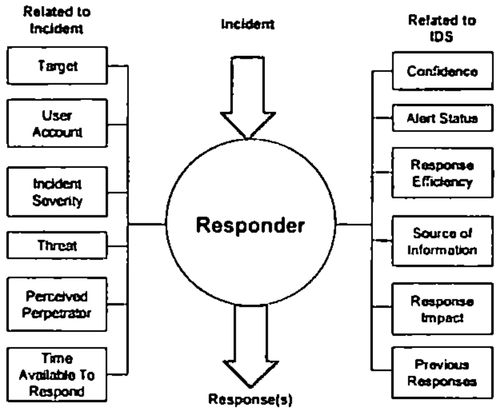


Figure 1: Contextual factors influencing intrusion response

The Intrusion Monitoring System (IMS)

IMS is a conceptual architecture for intrusion monitoring and activity supervision, based around the concept of a centralised host handling the monitoring of a number of networked client systems.

From the perspective of this research project, a significant element of the architecture is the Responder module, which is shown alongside other relevant entities in Figure 2 below.

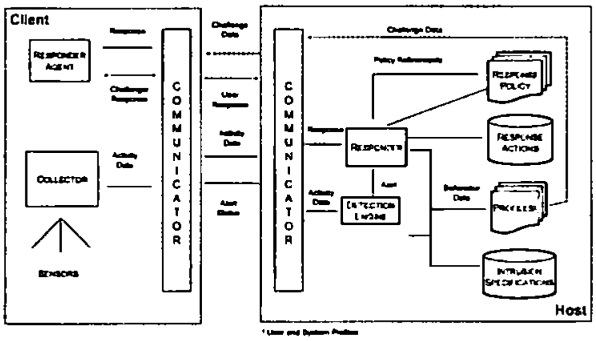


Figure 2: The Intrusion Monitoring System (IMS)

The **Responder** is responsible for monitoring the Alerts sent from the Detection Engine and, after considering them, in conjunction with other contextual factors, taking appropriate actions where necessary (e.g. correcting vulnerabilities, updating software, issuing authentication challenges, limiting access rights and increasing the monitoring level).

If the actions selected by the Responder need to be performed on the client side, a local **Responder Agent** is responsible for initiating and managing the process.

Having gathered all of the available information, the actions that should be initiated in different contexts are then specified in the **Response Policy**.

A Prototype Responder System

A prototype is being implemented to demonstrate the main response features of IMS, including the ability to make decisions based on the information from IDS alerts and other contextual factors. In the absence of a full Detection Engine, a console interface is provided to enable simulated intrusion alerts to be directed towards the Responder (see Figure 3).

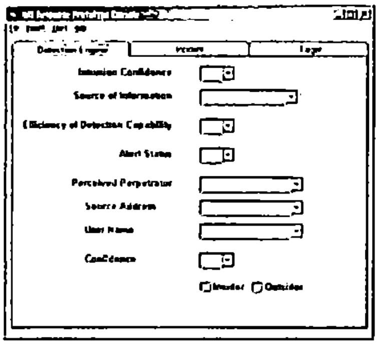


Figure 3 : Prototype Console Interface

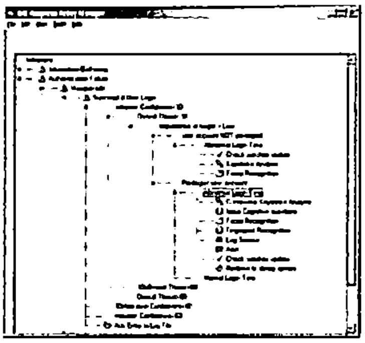
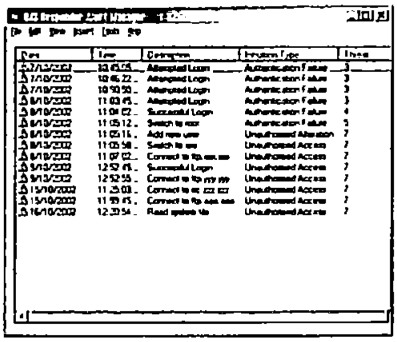


Figure 4 : IMS Response Policy Manager



Date	Time	Description	Intrusion Type	Threat
7/10/2002	10:45:25 ...	Attempted Login	Authentication Failure	3
7/10/2002	10:46:22 ...	Attempted Login	Authentication Failure	3
7/10/2002	10:50:50 ...	Attempted Login	Authentication Failure	3
8/10/2002	11:03:45 ...	Attempted Login	Authentication Failure	3
8/10/2002	11:04:02 ...	Successful Login	Authentication Failure	4
8/10/2002	11:05:12 ...	Switch to root	Authentication Failure	5
8/10/2002	11:05:16 ...	Add new user	Unauthorised Alteration	7
8/10/2002	11:05:58 ...	Switch to user	Unauthorised Access	7
8/10/2002	11:07:02 ...	Connect to ftp.xxx.xxx	Unauthorised Access	7
9/10/2002	12:52:45 ...	Successful Login	Unauthorised Access	7
9/10/2002	12:52:55 ...	Connect to ftp.yyy.yyy	Unauthorised Access	7
15/10/2002	11:25:03 ...	Connect to [illegible].zzz.zzz	Unauthorised Access	7
15/10/2002	11:55:45 ...	Connect to ftp.aaa.aaa	Unauthorised Access	7
16/10/2002	12:20:54 ...	Read system file	Unauthorised Access	7

Figure 5 : IMS Responder Alert Manager

The Responder largely bases its decision upon the Response Policy, which can be accessed by selecting the **Response Policy Manager** tool, illustrated in Figure 4.

At the most basic level, there will be a one to one correspondence between a type of incident and an associated type of response. However, a more likely situation is that the desired response(s) to an incident will vary, depending upon other contextual factors.

During normal operation, the Responder logs the details of responses that have been issued so that they can be tracked and reviewed by a system administrator. This is achieved via the Alert Manager interface (see Figure 5), which contains a list of suspected incidents, allowing them to be selected and reveal the response action(s) initiated for them.

Papadaki M, Furnell S.M, Lee S.J, Lines B.M, and Reynolds P.L. 2002. "Enhancing Response in intrusion detection systems", *Journal of Information Warfare*, Volume 2, Issue 1, pp90-102.
Papadaki M, Furnell S.M, Lines B.M, and Reynolds P.L. 2002. "A Response-Oriented Taxonomy of IT System Intrusions", *Proceedings of Euromedia 2002*, M.Roccetti (ed.), Modena, Italy, 15-17 April 2002, pp87-95.
Dowland P.S, Furnell S.M, and Papadaki M. 2002. "Keystroke Analysis as a Method of Advanced User Authentication and Response", *Security in the Information Society: Visions and Perspectives*, M.A.Ghonaimy et al (eds), pp215-226.
Irakleous I, Furnell S.M, Dowland P.S, and Papadaki M. 2002. "An experimental comparison of secret-based user authentication technologies", *Information Management & Computer Security*, vol. 10, no. 3, pp100-108
Papadaki M, Magklaras G, Furnell S.M, and Alayed A. 2001. "Security Vulnerabilities and System Intrusions – The need for Automatic Response Frameworks", *Proceedings of IFIP 8th Annual Working Conference on Information Security Management & Small Systems Security*, Las Vegas, 27-28 September 2001.
Furnell S.M, Magklaras G.B, Papadaki M, and Dowland P.S. 2001. "A Generic Taxonomy for Intrusion Specification and Response", *Proceedings of Euromedia 2001*, Valencia, Spain, 18-20 April 2001, pp125-131.

ΙΔΡΥΜΑ ΚΡΑΤΙΚΩΝ ΥΠΟΤΡΟΦΙΩΝ
IKY

www.network-research-group.org

Factors Influencing Automated Intrusion Response

Maria Papadaki[†], Network Research Group, University of Plymouth, UK

[†]This work has been funded by the State Scholarships Foundation of Greece

Abstract

With rising levels of attacks and misuse, intrusion detection systems are an increasingly important security technology for IT environments. However, while intrusion detection has been the focus of significant research, the issue of response has received relatively little attention. The majority of systems focus response efforts towards passive methods, which serve to notify and warn, but cannot prevent or contain an intrusion. Where more active responses are available, they typically rely upon manual initiation. The research to be presented is based on the Intrusion Monitoring System (IMS) Architecture and argues that a more comprehensive and reliable response framework is required in order to facilitate further automation of active responses. A range of factors are identified that a software-based responder agent could assess in order to improve response selection, and thereby increase trust in automated solutions.

Introduction

An increasing level of attacks upon IT systems represents a seemingly unavoidable reality of the Internet revolution. From the malicious activities of external hackers to deliberate misuse by organisational insiders, no sector has shown itself to be immune from attack. Evidence of the problem is provided by results from the annual CSI/FBI Computer Crime and Security Survey, in which the financial losses in 2002 have reached the level of $456 millions, an increment of 20% compared to 2001 and 355% compared to 1997.

Statistics such as those above emphasize the need for security in networked systems, and a key technique for combating attacks is provided by Intrusion Detection Systems (IDS).

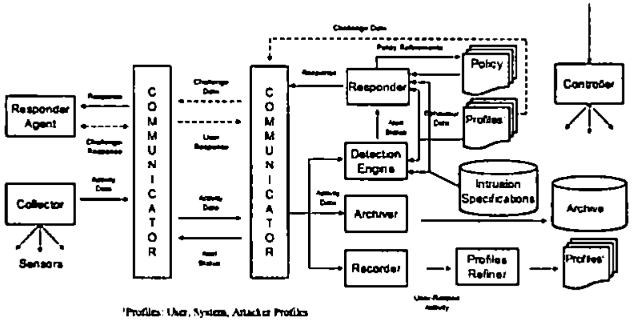


Figure 1: Intrusion Monitoring System Architecture

Intrusion Monitoring System (IMS) Architecture

The Intrusion Monitoring System (IMS) is the focus of security research in the Network Research Group. IMS is an architecture for intrusion monitoring and activity supervision, based around the concept of a centralised host handling the monitoring of a number of networked client systems. Intrusion detection is based upon the comparison of current user activity against both historical profiles of 'normal' behaviour for legitimate users and intrusion specifications of recognised attack patterns. The architecture is comprised of a number of functional modules, addressing data collection and response on the client side and data analysis and recording at the host (as illustrated in the figure above).

The module of the **Responder** is responsible for monitoring the Alerts sent from the Detection Engine and taking appropriate actions where necessary, based on the given context.

Forms of Response

Intrusion response can be defined as the process of counteracting the effects of an intrusion. In the context of intrusion detection it includes the series of actions taken by an IDS following the detection of a security-related event. The response options currently available can be categorised into two main categories: Passive and Active Responses.

Passive

Passive responses aim to notify other parties about the occurrence of an incident, relying on them to take further action. Passive responses may include actions such as:

- Adding an entry in a log file
- Alerting an administrator (e.g. via on-screen message, email, pager or SMS)
- Generating alarms and alerts to a network management console

Passive responses, have been used by IDSs since their very early stages and are still present in every intrusion detection product, offering the standard level of response in commercial IDS systems.

Active

Active responses are the actions taken to counter the incident that has occurred. Such actions might include the following approaches:

- Collecting more information (e.g. via an authentication challenge or increased monitoring)
- Limiting permitted user behaviour or process activity;
- Blocking traffic through firewalls and routers;
- Terminating network connections;
- Introducing delay on network connections.

Active responses can have more significant impact upon a system, and thus engender the danger of causing unwanted effects if falsely initiated.

The problem of Automation

Automated response can dramatically reduce administrative load. It is also suitable for combating automated attacks.

On the other hand, the automation of active responses can affect legitimate users in a false rejection scenario, and could enable the IDS to become an attack launching tool (if tricked by an attacker to issue active responses).

Factors Influencing Response

In order to address the automated response issue, it is important that further attention is given in the identification and assessment of factors that influence the response decision process.

The figure below depicts the main factors, split according to whether they relate to the incident or the IDS. The *incident* is the trigger for the response and still represents the principal influence over what should be done. However, assessment of the other factors enables the responder to establish the context of the incident, and select appropriate responses accordingly.

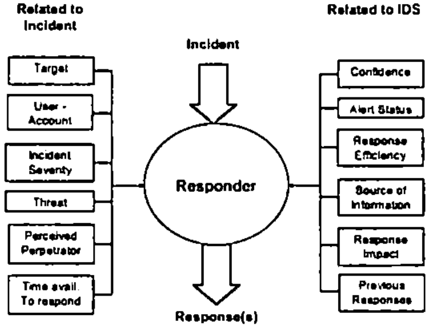


Figure 2: Contextual factors influencing intrusion response

Conclusion

In order to increase confidence in automated response systems, the decision making process underpinning response selection must be enhanced. This poster has summarised a range of factors that can influence the decision process. However, a deeper analysis is required to determine the relative importance of factors in different scenarios, and their inter-relationships. These aspects represent ongoing research, and further findings will be documented in later publications.

Further Information

M.Papadaki, S.M.Furnell, S.J.Lee, B.M.Lines, P.L.Reynolds, *"Enhancing response in intrusion detection systems"*, to appear in Journal of Information Warfare.

P.S.Dowland, S.M.Furnell, M.Papadaki, *"Keystroke Analysis as a Method of Advanced User Authentication and Response"*, Proceedings of IFIP/SEC 2002 - 17th International Conference on Information Security, Cairo, 7-9 May 2002.

University of Plymouth
Library

Item No 9006421885

Shelfmark
THESIS 005 . 8 PAP

COPYRIGHT STATEMENT

This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without the author's prior consent.

University of Plymouth
Library
Item No 9006421885
Shelfmark
THESIS 005.8 PAP

COPYRIGHT STATEMENT

This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without the author's prior consent.

Figures

Tables

Author's Declaration

At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other University award.

This study was financed with the aid of a three year CASE studentship from the Engineering and Physical Sciences Research Council (EPSRC), sponsored by OPD Management Consultants Limited.

The following publication was produced from the research:

Dixon, M; Jagodzinski, P; Pearce, P (1998) Software Support for Management Consultants: An Integrated Quantitative and Qualitative Tool. Proceedings of Seventh International Conference on Management of Technology, 16-20 February 1998. Orlando, Florida, USA.

Signature: M Dixon

Date: 12/07/2000

Acknowledgement

Many people have given assistance and support over the past six years, without which this work would have not been possible. I would like to thank my director of studies Dr. Peter Jagodzinski for his advice and guidance; Mr Dennis Clarke for his advice, support, and time; Mr. Brian Mushens for his advice regarding the field of Information Retrieval; Prof. Patricia Pearce for initiating the research studentship; all of the technicians within the University of Plymouth for their assistance with computer, audio, and video equipment; Mr Nathan Tillett for his illustrative skills; and my family and friends for their support and understanding.

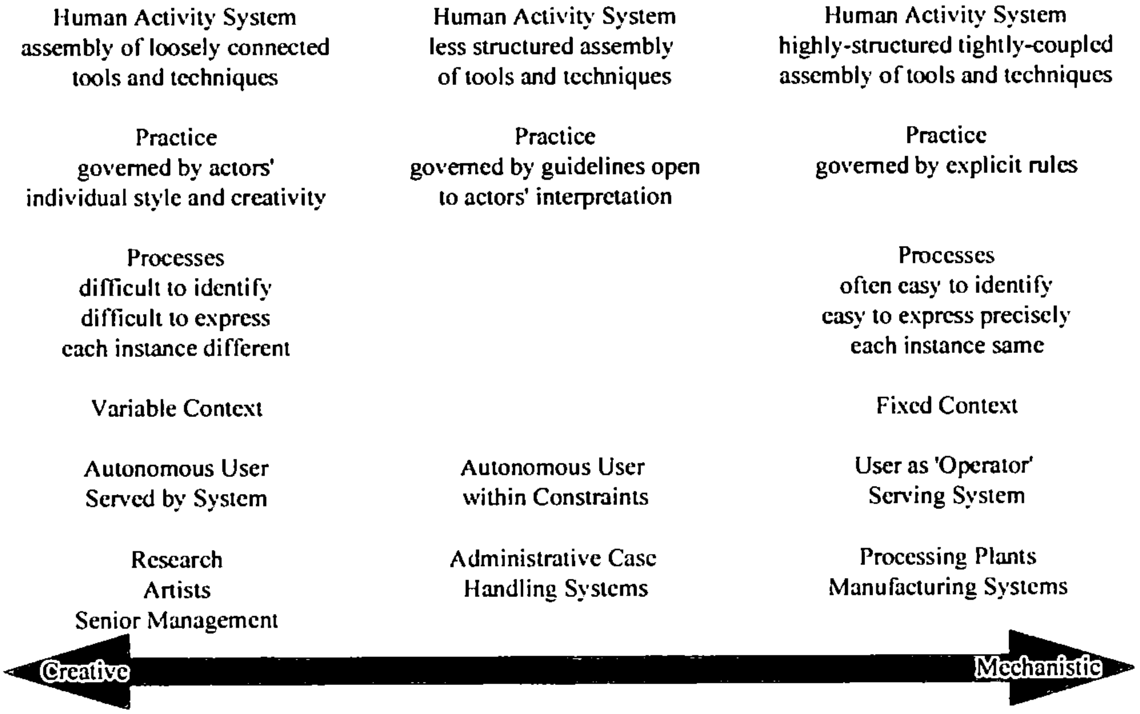
I would also like to thank the following organisations: OPD Management Consultants Limited for their support and sponsorship; EPSRC for their sponsorship; and the University of Plymouth.

1. Introduction

1.1 New Application Domains and New Methods: User Centred Methods for the Support of Creative Human Activity Systems

Over the past few decades the world has witnessed a rapid transformation of the power of computer hardware and reductions in its cost. As a result the use of computers is spreading into new application domains focused on creative human activity (Landauer, 1995; Shneiderman, 1998; and Shneiderman, 1999).

Fundamental differences between application domains have been identified and described (Flynn, 1992; Rasmussen, 1992; Shneiderman, 1992; and Landauer, 1995). For the purposes of the present work application domains may be placed on a continuum ranging from mechanistic to creative as shown in Figure 1.1.



Human Activity System assembly of loosely connected tools and techniques	Human Activity System less structured assembly of tools and techniques	Human Activity System highly-structured tightly-coupled assembly of tools and techniques
Practice governed by actors' individual style and creativity	Practice governed by guidelines open to actors' interpretation	Practice governed by explicit rules
Processes difficult to identify difficult to express each instance different		Processes often easy to identify easy to express precisely each instance same
Variable Context		Fixed Context
Autonomous User Served by System	Autonomous User within Constraints	User as 'Operator' Serving System
Research Artists Senior Management	Administrative Case Handling Systems	Processing Plants Manufacturing Systems

Figure 1.1 – Application domains continuum (after Rasmussen, 1992: p. 9).

The mechanistic type of application domain (referred to as phase 1 by Landauer, 1995) is typified by the simple transaction processing application domains (such as invoicing and pay roll), where the processes can be expressed explicitly and precisely using mathematical and algorithmic representations, and processes follow a prescribed pattern of activity. The structured software development methods that emerged for such application domains were appropriately reductionistic and rigid.

In contrast, the creative type of application domain (referred to as phase 2 by Landauer, 1995) is more complex where human creativity, intuition, individual experience, and personal judgement are fundamental to the processes. As a result these processes are highly variable (both between people and instances of application) and cannot be expressed explicitly and precisely. For example an artist does not follow a set sequence of actions every time he or she paints a picture, there are significant structural variations in the processes used each time. This type of creative human activity system has recently been successfully supported by user driven software that consists of a set of related facilities that the user can select at will, rather than a prescribed sequence of activities over which the user has little or no control.

However, in some areas problems have been experienced due to established software development methodology being unable to successfully determine user requirements. The software that has been developed to support these types of application domain has had poor usability: there has been a mismatch between the facilities that the software has provided and those that are required (illustrated by the cartoon presented in Figure 1.2).

'Sorry, Ladies and Gentlemen – it's not what you ordered, but everyone is getting fettuccine until we sort out the computer'

Figure 1.2 – Cartoon illustrating poor usefulness of software.

In response to this, new software development methods and techniques have been developed, such as user-centred design, soft systems methods, and knowledge engineering. Also, the potential contribution from social science disciplines (such as ethnography) has started to be assessed.

1.2 Management Consultancy as a Creative Human Activity System

1.2.1 Introduction to Management Consultancy

The services of consultants are frequently employed by organisations seeking to improve their commercial competitiveness, which frequently involves analysing a broad range of issues with a view to identifying and solving problems (Kubr, 1986). Consultants may be brought in to provide specialist knowledge and skill, an impartial outside view (Tisdall, 1982, and Kubr, 1986), gather evidence to aid the decision making process, or perform long term (non-operational) work. They may also be brought in to identify good practice in order to transfer it to other areas of an organisation, and are often closely associated with the diffusion of new methods and technologies. It requires creative thinking, imagination, and interpersonal skills on the part of the consultant (Kubr, 1986; Markham, 1994; and Margerison, 1988). Many management consultancy companies follow a non-prescriptive style that involves a high level of participation from the client organisation.

A significant problem facing consultants is the volume of information available for collection in an organisation, and that a large amount of information easily becomes incomprehensible and cannot be fully utilised (Kubr, 1986). This is often referred to generally as the problem of 'information overload'. The effect of this is for the consultant to restrict the amount of information gathered, at the risk of missing vital information. Kubr (1986: 11) also indicates that organisations may 'find such professional service too expensive'. The effect of this is that management consultancy services are beyond the budget of many (especially smaller) organisations.

The process of consultancy is very sensitive and as a result surrounded by secrecy (Hyman, 1961; Tisdall, 1982; Rassam and Oates, 1991; and Czerniawaska, 1999): it often involves issues that are delicate, important, controversial, personal, emotive, and confidential, which consultants are usually unwilling to discuss with outsiders.

1.2.2 Software Tools

The importance of creative thinking for management consultants suggests that it lies toward the creative (left hand side) of the application domain continuum. However, the more precise placement of management consultancy on the continuum depends on understanding the nature of tacit knowledge regarding exactly how consultants work (management consultancy methodology). A detailed description of this tacit knowledge is not readily available in the management consultancy literature, which is not surprising as its experience based, individualistic, and confidential nature make it difficult for consultants to articulate.

Recent research has worked to provide software tools that begin to address the problems faced by management consultants, with much of it aimed at reducing the cost (Baligh, *et al* 1996; Sushil and Raghunathan 1994; Krallmann *et al*, 1992; Franz and Foster, 1992; and Frowein and Postma, 1992).

HUSAT (1988) reports that, in general, basing the design of software on user requirements, carefully elicited through participation, increases software effectiveness. However, the research literature relating to management consultancy software does not contain an explicit user model of the management consultancy process on which to base software development.

Instead much of the existing research has been driven by domain specific theory (such as organisation theory), and technology (especially expert systems technology) rather than being led by users' requirements. For example, it is possible to represent certain domain knowledge (for example in rule bases) so that it is computable: organisational domain knowledge is represented as production rules to analyse specific aspects of the collected organisational information, and generate prescriptive recommendations.

However, working in this prescriptive rule-based manner does not match the non-prescriptive, creative practices of many management consultants: the software is placed on the mechanistic (right hand side) of the application domain continuum, whereas much of management consultancy appear to be placed toward the creative (left hand side).

This is not to suggest that the use of expert systems technology or organisation theory is inappropriate. Rather that the effectiveness of software has been limited due to user characteristics and practices not being a focal factor in their development. It also follows that the effectiveness of software would be significantly enhanced by consideration of the consultant's (user's) requirements, which may involve the use of other technology, and knowledge from other disciplines.

It is possible to distinguish between two approaches to software development that have been taken:

- Much of the software has been designed for use by in-house managers, as a replacement for consulting services. This form of software will be referred to as computer based management consultancy (CBMC) within this thesis. With this form of package, managers within client organisations are the target users, and the aim is to provide managers with expertise regarding specific areas: such as Total Quality Management (TQM).
- Some of the software has been designed for use by consultants. This form of software will be referred to as computer aided management consultancy (CAMC), derived from the term computer aided consultancy (CAC) used by Krallmann, *et al* (1992: 264). With this form of package, the consultants themselves are the target users, and the aim is to support their practices.

Although replacing the management consultant with CBMC software can reduce cost, certain services can only be provided by consultants themselves: consultants can provide services that software alone cannot, such as:

- experience and creativity in the form of subtle tacit judgements
- analysing a broad range of issues,
- providing an impartial outside view, and
- using their individual interpersonal skills to elicit information that individuals in an organisation would be unwilling to supply to either management or a computer.

It is therefore important that more work is undertaken in the field of CAMC to support the practices of management consultants. Thus, the present work addresses the development of CAMC software tools.

The management consultancy process presents some interesting challenges to the development of software support. At the start of the present work, very little was known by the author about the problem domain and the tasks that the software would support. This and the limited details of how consultants work in the literature highlighted the need for user-centred methods, especially the participation of experienced management consultants, in the software development process. Fortunately, a particular consultancy company was interested in the work, and consequently its members were willing to make themselves available for a user-centred study.

The present work seeks to employ, investigate, and augment user-centred methods to tease out the characteristics of management consultancy and hence place the field more precisely on the application domain continuum.

The hypothesis of the present work is that *augmented user-centred methods can reveal previously unidentified, fundamental user requirements in creative application domains, such as management consultancy.*

1.3 Research Aims

The domain of the present work was the investigation of the effectiveness of user centred approaches to the development of software tools that support creative human activity systems (HAS), such as the analysis of organisational issues by management consultants.

This was the setting for the research aims, which can be seen at three levels:

- To investigate the development of improved software tools for management consultants.
- To investigate the effectiveness of user centred methods in the development of software support for management consultants, and augment such methods where necessary.

- A broader intention of the research was to consider the more general use of user-centred methods to develop software tools that support human activity systems at the creative (left hand side) of the application domain continuum.

In this way the work produces a contribution to knowledge (described in section 7.1).

1.4 Research Objectives

The objectives of the research programme were to:

- develop a user model of the management consultancy process
- identify and evaluate the potential contribution to management consultants of software developed for use in other related disciplines
- determine the current state of software support for management consultants and evaluate it against the above mentioned user model of the management consultancy process
- develop a generic design model of software support for the management consultancy process
- evaluate the potential impact of software tools based on that model. This would include consideration of advantages, disadvantages, and changes to the management consultancy process that may occur as a result of the use of such software tools.
- Evaluate the effectiveness of user-centred methods throughout the system development lifecycle
- Review the role of user-centred methods in the design of software tools to support creative human activity systems

Specifically this meant the development of a prototype software tool for management consultants as an executable specification, used to clarify, and then to empirically evaluate the model.

Its design was to be based on the consultant's mental model of the consultancy process, elicited by user-centred methods, and to seek to enhance the ability of management consultants to maintain a richer mental model of complex organisations and hence understand them better, rather than simply automating the mechanical data processing aspects of the management consultant's role.

This work is focused on the effectiveness of the software in enhancing the ability of management consultants to comprehend complex organisations, rather than increases in efficiency through optimising the speed of automated mechanical data processing aspects of the management consultants' practices.

1.5 Research Scope

The work addressed the support of the consultancy process itself and not the administrative activities that support it (such as accountancy, project management, secretarial) as the software support (often in the form of standard packages) for these activities is relatively good when compared to the consultancy process itself. The research programme focused on management consultants:

- performing organisational analysis,
- working individually,
- following a non-prescriptive approach, and
- working across a range of domains.

1.6 Nature of the Research

The overall aim of the present work focuses on solving a 'real world' problem (i.e. management consultancy). Such research is referred to as 'problem solving' research by Phillips and Pugh (1994: 50). As the development of user requirements for CAMC is a relatively new field about which little is known the present work has also been 'exploratory' in Phillips and Pugh's (1994: 49) terms.

The present work is multi-disciplinary, involving the industrial field of management consultancy, the analysis disciplines of social network analysis (SNA) and qualitative data analysis (QDA), and computing disciplines including information retrieval (IR), relational database systems (RDS), and user centred design (UCD).

1.7 Overview of Thesis

This section provides an overview of the thesis by summarising the content and structure of its chapters. It identifies their aims, and indicates where contributions to knowledge are claimed. One contribution to knowledge of this programme of research has been within the field of computer aided management consultancy (CAMC). A second area of contribution has emerged in the field of user-centred systems design.

The thesis has several literature reviews distributed throughout its chapters, rather than a single literature review chapter. This was due to the multi-disciplinary nature of the work, and has been done to present literature reviews in context with other related material.

It is important that any interpretation by the researcher is visible and hence open to examination. To this end the results (a faithful descriptive summary of data from various sources) and conclusions (interpretations relating material from several sources) are presented separately.

Chapter 1 argues that advances in computing have led to computers being used in new application domains, which involve human creativity, such as management consultancy. However, problems have been encountered in some areas. This has resulted in new software development methods emerging.

Chapter 2 argues that a combination of new and old methods should be used to develop software for creative application domains, such as management consultancy. It presents an overview of the research methods used, and the rationale for their use. The overall approach taken was user-centred, participative, and holistic.

Chapter 3 presents a user model of management consultancy that describes characteristics and practices of management consultants, which identifies that it is based on human creativity, experience, intuition, and integrated quantitative and qualitative analysis. This provides empirical case study and literature based evidence that management consultancy is an example of a creative application domain.

Chapter 4 argues that integrated social network analysis and qualitative data analysis functionality will be useful to management consultants. In doing so it describes concepts, methods, and software tools from the disciplines of Organisational Behaviour (OB), Social Network Analysis (SNA), and Qualitative Data Analysis (QDA) that were identified as being similar to those described in the user model of management consultancy (presented in Chapter 3), and which might therefore contribute to the development of software tools for management consultants.

Chapter 5 describes a generic design model for computer assisted management consultancy (CAMC) software tools, which incorporates a design rationale and functional specification. This design model was based upon the user model of management consultancy (described in chapter 3); an integration of current SNA, and QDA concepts, methods, and software support (described in chapter 4); and current CAMC software tools (described within this chapter). It includes the description of a prototype Integrated Social Network and Qualitative Data Analysis (ISNQDA) software tool that was developed as a vehicle for evaluating the design model, and the impact of using such software on the consultancy process.

Chapter 6 provides empirical evidence that demonstrates that CAMC software tools based on this model can be useful to management consultants, and that the use of such software tools would change the consultancy process. In doing so it describes the results of two holistic evaluations: an empirical user test with the prototype software tool (described in chapter 5), and a user acceptability interview.

Chapter 7 summarises the findings of the above chapters; reflects on the effectiveness of employing user-centred methods in the development of software tools for a creative application domain, such as management consultancy; and discusses possible future work. In doing so it identifies and describes in detail the contribution to knowledge made by the present work.

2. Research Method

This chapter presents a description of and rationale for the approach taken and methods used by the present work. It starts by giving an overview of current software development methodology, which describes conventional software development methods, their criticisms, and some of the new methods developed to address those criticisms (mentioned in Chapter 1). It then describes the methods used in the present work, the rationale for using them, and relates them to the current software development methodology.

2.1 Software Development Methodology

This section picks up from section 1.1 by giving a more detailed description of conventional software development methods (structured systems analysis and design methods, and software engineering), their criticisms, and some of the new methods developed to address those criticisms (soft systems methods, and user-centred methods). In doing so it outlines the software development methods that collectively formed the foundation of the research methods employed by the present work.

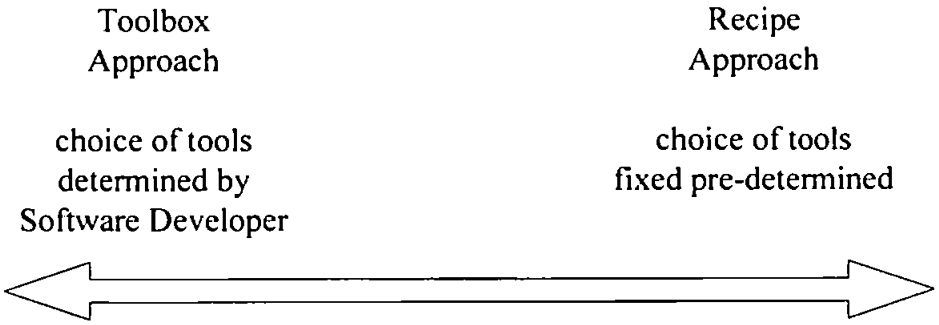

Toolbox Approach
choice of tools determined by Software Developer
Recipe Approach
choice of tools fixed pre-determined

Figure 2.1 – The Software Development Philosophy Continuum (after Crinnion, 1991: p. 14).

The transformation that has taken place in software development methodology can be described as a movement from the traditional structured approach, which sought a single method with prescribed phases and stages, toward the broad range of philosophies, strategies, methods, and techniques that exist today and allow the software developer to select those most appropriate to the problem domain at hand (shown in Figure 2.1). The left hand 'toolbox' approach is generally more applicable to software development in 'creative' application domains, whilst the right hand 'recipe' approach is more applicable to 'mechanistic' application domains.

Each of the following sections describes areas of software development methodology; identifying the problems it addresses and its limitations.

2.1.1 Structured Systems Analysis and Design Methods, and Software Engineering

Structured systems analysis and design methods (also referred to as hard, traditional, conventional systems analysis and design methods) and software engineering methods are the foundation of modern systems analysis. All of the approaches mentioned in later sections are regarded as complementary to rather than replacements for structured techniques.

They were developed in response to the problems resulting from the ad-hoc, 'build it – fix it' software development style of the 1960's and 1970's, which focused heavily on implementation, included few management and control mechanisms, and little or no contact with or consideration of the user. This frequently resulted in users being dissatisfied with the systems that were built; as they frequently did not fulfil the user's expectations, were unreliable, were delivered late, and went over budget. Structured systems analysis and software engineering methods recognised the importance of analysis and design activities, user involvement, and the management and control of the development process.

The fundamental basis of structured methods and software engineering is function decomposition: that is breaking a complex problem down into smaller constituent parts, thus simplifying it and making it more manageable (Avison and Fitzgerald, 1995).

A well defined, sequential set of phases and stages were developed to describe the life cycle of software development projects that were well structured and systematic; resulting in the classic 'waterfall model' of the software development life cycle, shown in Figure 2.2. This allowed standard project management and control techniques to be applied to the software development process; identifying deliverables and milestones, and estimating time-scales.

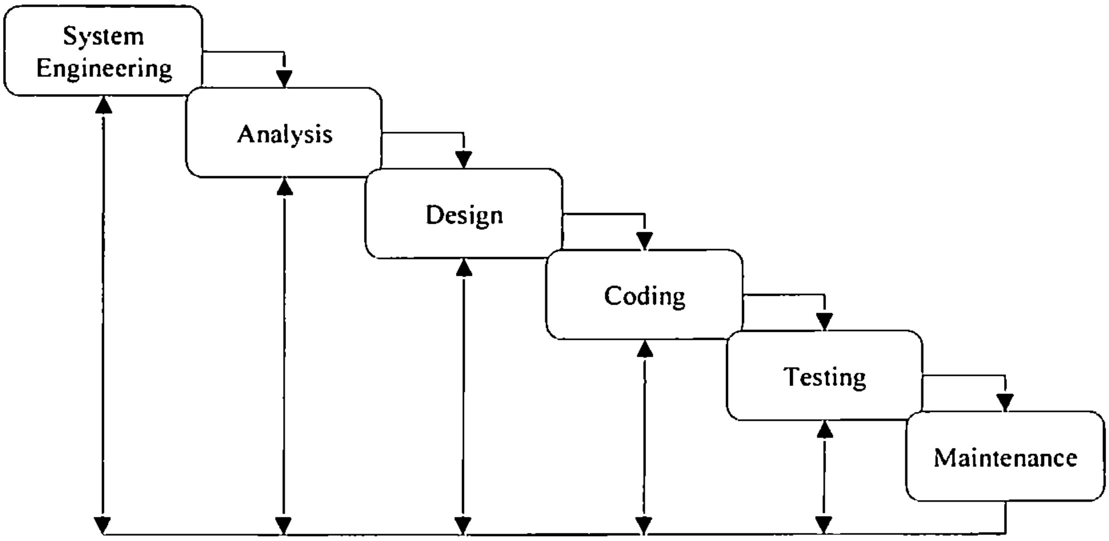


Figure 2.2 – The classic ' waterfall model' software development life cycle (Pressman, 1992: p. 25).

An effort was made to separate functionality (what the software should provide) from implementation (how this was achieved), which led to the idea of the software requirements specification (Pressman, 1992; and Sommerville, 1996): a document that explicitly, precisely, and unambiguously described what the user required the system to do without regard or mention of how it would be done. This document could then be used as a benchmark for development, sometimes acting as a contract between the developers and the user.

Modular design (Pressman, 1992; and Sommerville, 1996) was used to divide software into separately named and identifiable components (referred to as modules) that are integrated to meet the software requirements specification. This reduced software complexity, and made modifications faster and easier as parts of the system could be developed in parallel with less chance of unexpected interactions (often called side effects).

Rigorous testing methods were developed, such as regression testing (Pressman, 1992; and Sommerville, 1996), which advocated repeating tests that had previously been passed, to detect new errors that may have been introduced as a result of recent modifications; this dramatically increased the reliability of software.

Several information gathering techniques were used (Avison and Shah, 1997; Bingham and Davies, 1992; and Flynn, 1992). The analysis of documentation that exists within a user's organisation is frequently used to gather information. Interviews are another frequently used information gathering technique. A significant problem with the latter method is that people often find it difficult to express their actions and behaviour after the event. What often results is a post-hoc rationalisation of their behaviour, which may not accurately describe what they actually do. Observation of the user's behaviour in their working environment is an established systems analysis technique for gathering information. The main limitations of this method are that the act of observing people may change the way they behave, and that it does not disclose the tacit knowledge that the user is applying.

A wide range of techniques are used to represent information that is collected. These tend to be based on structuring the data and looking at it from different perspectives. Common techniques are data flow diagrams (DFDs), which model processes and data stores, and entity-relationship diagrams (E-R diagrams), which model data as entities and the relationship between them.

2.1.1.1 Criticisms

These methods were successful in addressing some of the problems of software development. However, as computers became more powerful and were consequently applied to different application domains, new problems emerged. Four of the major criticisms are:

- That they are technology led, focusing on the solution rather than the problem (Clegg, Waterson, and Carey, 1994, and King and Majchraz, 1996), which can result in inappropriate technological solutions being applied due to misunderstanding the problem.
- That, due to their reductionist nature, contextual and human factors issues (social and organisational issues) are given little consideration and treated in a mechanical and deterministic manner (Avison and Fitzgerald, 1995).
- That they put a large effort into developing a precise and unambiguous software requirements specification with executable software being produced during the latter stages. It has been established that it is often very difficult, if not impossible for precise and unambiguous software requirements specifications to be elicited. This leads to an inherent risk that with no opportunity for users to try out the software until almost the end of the project errors in the specification will only be detected at the end of the project when they are difficult, time consuming, and costly to correct (Neilson, 1993).
- That they assume that requirements are static, i.e. do not change over time; evidence has shown that this is not the case and as a result systems have been developed that are of little use as by the time they are implemented the user's requirements have changed (Crinnion, 1991).

The following sections describe soft systems methods and user-centred system design methods, which were developed in response to these criticisms.

2.1.2 Soft Systems Method s

Soft systems methods (SSM), described by Checkland (1981), have been more recently applied to improve and deepen the analyst's understanding of the problem situation rather than the solution. This strongly distinguishes SSM from structured methods (described in the previous section) that are criticised for focusing too heavily on the solution before the problem is adequately understood. It is important to note that soft systems methods are not (and were never intended) as replacements for structured methods, and, in fact, naturally complement them: typically soft methods would be used first to understand the problem, and would be followed by hard methods to develop a solution.

Soft systems methods are especially useful in problem domains that are 'complex and cluttered by people and their individual perceptions' (Patching, 1990: p. 33): they facilitate reconciliation of different perspectives of the problem, and recognise the importance of the context of the problem.

The term 'holon' (Checkland and Scholes, 1990) is used to refer to the abstract idea of a perspective (created by a person) of some part of the world that may be considered as a whole. In this way SSM recognises the existence of many different perspectives of the world. Each perspective is restrictive and incomplete, and debate and discussion among those who hold them will result in a more comprehensive overall understanding of the problem situation (Ho and Sculli, 1994).

The focal form of the holon is the human activity system (HAS), described by Wilson (1990), which has several important characteristics: they consist of humans engaged in purposeful activity. Human activity systems are open systems – they do not exist in isolation: they are affected by an external context. In this way a holistic perspective

(Patching, 1990) is taken; analysis is not limited to a narrowly focused preconceived area of interest, the broader picture is taken into consideration.

A key technique within soft systems methods is building a rich picture, which seeks to represent a deep and holistic understanding of the problem. Rich pictures usually take a diagrammatic form, however this is not always the case.

Probably the most significant problem with soft systems methods is that much of the secondary literature has fundamentally misunderstood it (Checkland and Scholes, 1990), due to it involving a fundamentally different way of thinking. This makes it difficult to teach and has inhibited its widespread use.

2.1.3 User Centred Systems Design

Preece *et al* (1994) describe the overriding aim of user-centred methods (UCM) for systems analysis and design as the production of computer systems that are 'effective in facilitating the activities that people want to undertake' (p. 360). This is embodied in the four principles of user centred system design given by HUSAT (1988: p. 17):

- User's Purposes: All computer-based systems exist for human purposes.
- User's Goals: The criteria by which systems are to be judged are taken to be the requirements of those who are to use them.
- User's Involvement: Design teams must have appropriate user involvement in all stages of the system life cycle.
- User's Characteristics: Systems must be designed explicitly to take account of user's characteristics.

In this way user-centred methods seek to tackle the problem of computer systems failing to address user's needs (described in section 2.1.1.1). These needs may be considered in terms of three fundamental concepts: usefulness, which is concerned with whether the facilities provided are useful to the user; usability, which considers whether the facilities are easy to use in everyday work; and learnability, which addresses whether the facilities are easy to learn. Traditionally, the focus was on usability and learnability, however more recent methods and thinking are placing greater emphasis on usefulness, that is to say, the functional relevance of the system to its users.

Another key concept is the mental model, which is defined by Norman as 'the model people have of themselves, others, the environment, and the things with which they interact' largely based on experience (1988: p. 17), and has been developed to account for the variability observed in human cognitive activity (Preece *et al*, 1994).

The principal technique of user-centred methods is user participation: the idea that users should be involved in all stages of the software development life cycle, that those involved should be representative of the people who will actually use the computer system (Nielson, 1993). User participation occurs in two settings: analysis of user's characteristics and working practices and evaluation of computer systems. Landauer (1995) notes that successful projects employing user-centred methods have at least two of the following: analysis, formative evaluation (undertaken to guide changes to the computer system) and summative evaluation (undertaken to judge its worth).

2.1.3.1 User Requirements Analysis

The deliverable from this activity is a user requirements specification, which describes the user's needs in their own terms – the tasks they do, irrespective of the means of meeting those needs; that is, it does not consider technology, such as computer software. This is distinct from the software requirements specification, which is a description of what functionality will be provided by the software (as described in section 2.1.1).

Task analysis is a key activity in understanding users' working practices; it seeks to identify and represent the tasks users perform, which assists in designing systems that more accurately reflect what users want to do. Tasks are usually represented as a hierarchical decomposition of tasks into subtasks.

When attempting to understand user's characteristics and working practices the basic information gathering methods of interview, questionnaire, observation, and documentary records (from structured systems analysis and design) are used and augmented. Many authors (Shneiderman, 1998; Landauer, 1995; and Nielson, 1993) describe variations of a 'thinking aloud' approach, where analysts ask users to vocalise their thoughts as they work. The effect of this is that 'the context of actively performing the job brings things to mind that aren't thought of later or mentioned in an interview' (Landauer, 1995: p. 278). This also provides the opportunity to ask questions as issues emerge, while they are fresh in the mind of both the user and the analyst, and a concrete situation exists. A potential limitation of this technique is that it may interfere with the process being observed. Video recordings are sometimes made of these sessions for later review (Downton, 1991; and Landauer, 1995): however, this generates large volumes of data and is therefore very labour intensive. Also, the benefits of extensive recording and review as a supplement to live observation is not yet clear.

Ethnography

A more recent development within user-centred methods is interest in the use of methods drawn from the social sciences, such as Ethnography (Avison, 1997; Hughes, 1995; Hughes *et al*, 1994; Hughes *et al*, 1993; Sommerville *et al*, 1993; and Sommerville *et al*, 1992).

The field of ethnography is concerned with describing and understanding cultures from an internal perspective (as viewed by the members of those cultures). As such it is data driven (theory emerges from the data); it does not attempt to fit cultures into a prescribed theoretical framework. It adopts an approach that is naturalistic (the cultures are studied in their natural setting as far as possible) and longitudinal (studies are undertaken over long periods of time – one or more years is not uncommon). It represents an attempt to unearth deeply embedded tacit knowledge and implicit practices; to make visible the invisible aspects of what people do.

The use of ethnographic methods in the development of computer systems is a natural extension of the conventional observation methods (described in section 2.1.1), as such it is primarily concerned with requirements analysis. However, it takes these conventional observation methods much further in several ways. It provides a more formal methodological basis for observation with explicit principles and reasoning. This can be seen by considering that almost all of the existing systems analysis literature mentions observation, but allocates a maximum of a paragraph to its description; giving little guidance to how it should be performed. It also takes the idea of user participation further, and in fact reverses it so that the developer participates in the user's world.

The use of ethnographic methods has two main limitations. Firstly, arranging the intense and long term access to users and their time required can prove problematic, especially in creative application domains which are often high pressure and competitive in nature and where expertise is in short supply (Turban, 1993). Secondly, although ethnography has been employed in several projects (such as Hughes, 1995), there has been no recognised approach to integrating the diverse information produced into the software development process; it has been difficult for designers to understand the relevance of the information and hence effectively use it to inform design.

2.1.3.2 Prototyping and Evolutionary Systems Development

Prototyping address the problem of mistakes in the software requirements specification not being detected until late in the project lifecycle (described in section 2.1.1.1). The basic idea is that a prototype is developed, which represents a sub-set of the full system, that can be used by real users. The prototype is then exposed to evaluation sessions (as described in section 2.1.3.3), sometimes referred to as exercising the prototype (Crinnion, 1991). It is most useful in complex situations where the user has only a vague idea of what they require; in such cases it can act as an executable software requirements specification (Pressman, 1992). Evolutionary Systems Development is a term used to describe the situation where the 'prototype becomes the new system' (Crinnion, 1991: p. 25). It was developed to address the problem of users' requirements constantly changing (described in section 2.1.1.1). At first glance prototyping may appear to be a return to the 'build it – fix it approach': the difference is that prototyping is planned and controlled rather than ad-hoc.

Figure 2.3 contrasts the traditional software development life cycle, which is broken into phases with fixed boundaries, with the evolutionary systems development lifecycle, in which phases overlap considerably.

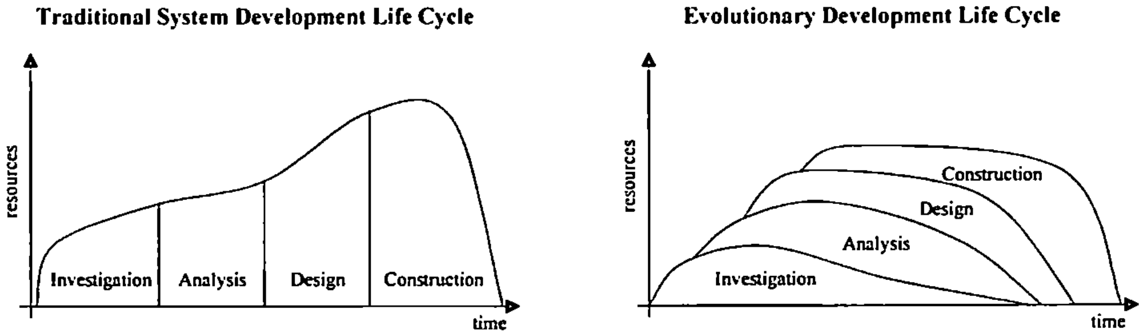


Figure 2.3 – Traditional and Evolutionary Development Life Cycles (Crinnion, 1991: p. 23).

Prototyping is dependent upon the prototype being fast and cheap to build. There are three ways in which this may be achieved: by limiting the facilities and functionality provided, by placing less emphasis on efficiency, and by using development tools specifically designed for prototyping.

Nielson (1993) distinguishes between horizontal and vertical prototypes, based on the 'two dimensions of prototyping' (p. 95): features and functionality (illustrated in Figure 2.4).

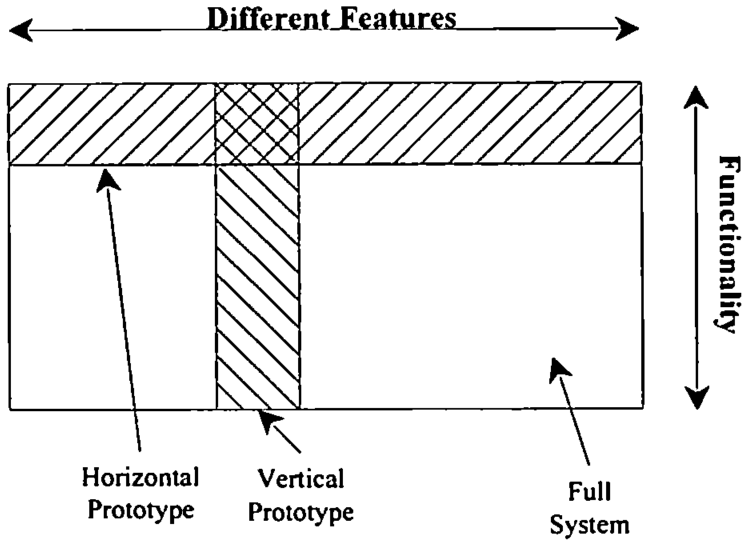


Figure 2.4 – The two dimensions of prototyping (Nielson, 1993: p. 95).

Horizontal prototyping reduces the level of functionality producing a system consisting of a superficial layer, which includes a broad range of features but provides no underlying functionality: effectively a simulation or facade. This allows a wide range of facilities to be evaluated, but is somewhat less realistic as users cannot perform real tasks. Vertical prototyping provides a limited number of fully functional features. This facilitates in-depth evaluation, but restricts the number of features that can be evaluated.

The prototype's cost and development time may be further reduced by less consideration of the efficiency of implementation: optimisation factors such as disk space, memory usage, reliability, maintainability, and execution speed may be given less resources than would be the case for a full system. However, Neilson (1993) notes that although test users may tolerate slow execution speed, if taken to an extreme it can cause frustration and make them over critical.

Several prototyping tools are used that allow software developers to make rapid and frequent changes to software without a loss in reliability and maintainability possible. These include fourth generation languages, database management systems, and screen generators.

When the prototype is completed it is important to distinguish between which aspects of it are intentional and which are arbitrary; otherwise the prototype may act as an over-specification – elements that were not explicitly designed may be incorporated into the full system.

2.1.3.3 Evaluation

The importance of empirical user testing, with real users, has been widely recognised; Landauer (1995: p. 281) describes it as 'the gold standard', and Nielson (1993: p. 165) describes it as 'the most fundamental usability method'. It provides direct concrete evidence of how they interact with the computer system. Two key methodological issues with user testing are reliability, which considers whether the tests are representative of the user population, and validity, which is concerned with whether the tests actually reflect the issues of interest. Reliability can be increased by increasing the number of evaluations and test users. Typical problems with validity result from atypical users, atypical tasks (such as 'toy tasks' that are an oversimplified and thus poor representation of reality), and/or an atypical setting (the choice between conducting testing in a laboratory or the users' place of work). Although use of the naturalistic setting may elicit more accurate information, the presence of the analyst may interfere with the setting, thereby distorting the results; and the requirements of the natural setting (such as a tight schedule) may inhibit the effective elicitation of information.

As identified above, the selection and number of test users is critical; Nielson (1993) identifies the main rule of selecting test users as that they should be as representative as possible of the actual people who will use the system. Ideally test users should be selected from several different sub-populations to cover the variety of people who will use the system. Where few users are available, they should at least be typical.

Nielson (1994) describes heuristic evaluation as a discount usability engineering method that can be used as part of an iterative design process, which is fast, cheap and easy to learn. It involves a small number of evaluators who examine the software and judge its compliance with recognised usability principles or heuristics. Empirical studies have indicated that increasing the number of evaluators to between three and five significantly improves the effectiveness of the method, as different people identify different issues. It is a good second best to user testing; however, it is important that the evaluators work independently of one another (Landauer, 1995).

Lastly, the user's subjective satisfaction (described by Shneiderman, 1998; Landauer, 1995; and Nielson, 1993) with the system may be elicited via interviews, which can reveal details of how they might behave toward the software. A more objective view may then be obtained by considering the responses of multiple users.

2.2 Overview of Resea rch Method of Present Work

This section describes the particular research methods used in the present work, which combined features and techniques drawn from structured systems analysis, software engineering, user centred methods, evolutionary systems development, and soft systems methods (described in the previous section).

User-centred methods were adopted as the main research paradigm of the present work, as by focusing on the effectiveness of software tools in assisting users they support the focus of the present work in considering the usefulness of CAMC software tools for management consultants. A soft systems approach was only adopted in part as the difficulty with consultancy was the invisibility of consultants' methods and practices rather than a clutter of multiple user perspectives, particularly as only a single consultant was involved.

An overview of the research method used in the present work, including the four major activities that were undertaken, is represented diagrammatically by Figure 2.5.

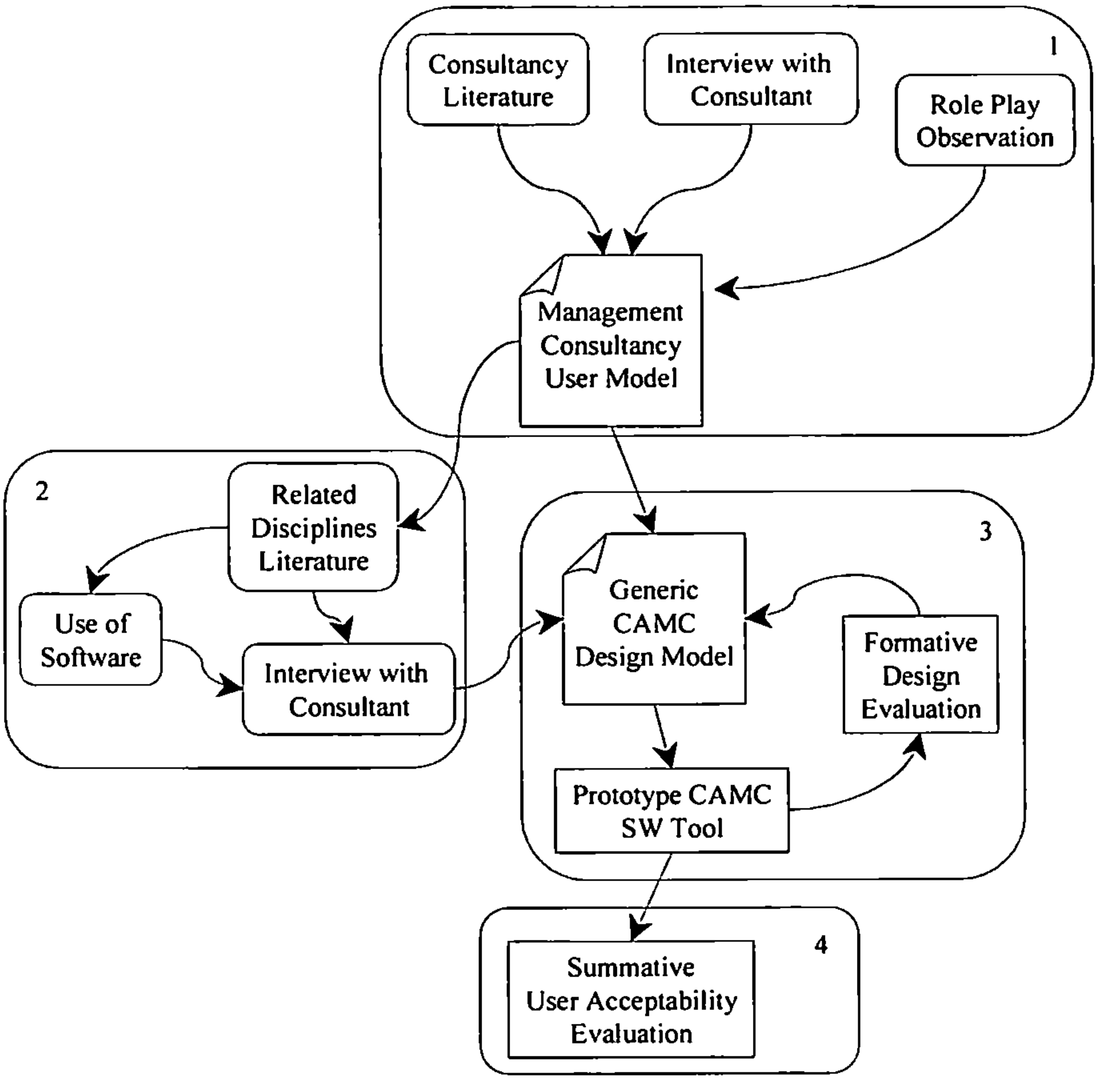

Consultancy Literature
Interview with Consultant
1
Role Play Observation
Management Consultancy User Model
2
Related Disciplines Literature
Use of Software
Interview with Consultant
3
Generic CAMC Design Model
Formative Design Evaluation
Prototype CAMC SW Tool
4
Summative User Acceptability Evaluation

Figure 2.5 – Overview of Research Method.

It includes the following phases:

1. User Model of the Management Consultancy Process (described in detail in Chapter 3): a general model of the user's characteristics, practices, and problems was developed, which provided a deep understanding or rich picture of the management consultancy process. This focused on understanding the problem and deliberately avoided considering solutions and technology.

2. Disciplines Related to Management Consultancy (described in detail in Chapter 4): other disciplines that were likely to contribute to the development of CAMC software were identified (based on similarity of methods with the user model of the management consultancy process) and investigated.

3. Generic Design Model for CAMC Software Tools (described in detail in Chapter 5): A generic model of CAMC Software Tools was developed, based on the user model, information from related disciplines, a review of existing CAMC software tools, and formative evaluations of a prototype CAMC software tool. The prototype was also developed as a vehicle for its evaluation in the next phase.
4. User Acceptance Evaluation (described in detail in Chapter 6): a summative evaluation of the generic model of CAMC software tools was conducted, via the prototype, which aimed to identify the impact of the use of CAMC software tools by management consultants: benefits, limitations, and resulting changes to the consultancy process.

They were performed as overlapping concurrent activities (as shown in Figure 2.6), rather than in a strict sequence. This is similar to the Evolutionary Systems Development Life Cycle (described in section 2.1.3.2).

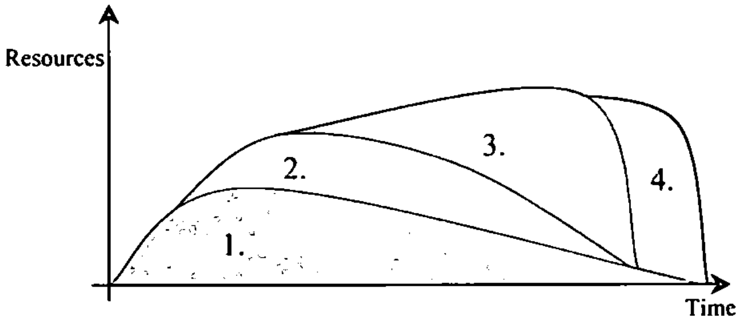


Figure 2.6 – Software Development Life Cycle Model.

The present work shows several features in particular:

- User Participation (described in section 2.1.3): A management consultant was at the centre of the present work, and participated in almost all aspects of it.
- Naturalistic investigation (described in section 2.1.3.1): An attempt was made to re-create as far as possible and investigate the users characteristics and practices in the natural setting.

- Holistic Analysis (described in section 2.1.2): An important concern regarding the current work was that CAMC software that provided benefits to one aspect of the consultancy process, could damage other aspects. Therefore a holistic view was taken; specifically, the analysis process (within consultancy assignments) was considered in context with other tasks and management consultancy as a whole.
- Structured Analysis (described in section 2.1.1): Structured analysis methods were used during the initial requirements analysis to describe data, processes, and data flows between processes involved in management consultancy assignments.
- Iterative Design (described in section 2.1.3.3): The design of the prototype followed an iterative process of formative evaluation and modification.

2.2.1 Deriving a User Mod el of the Management Consultancy Process

This activity (described in more detail in chapter 3) aimed to develop a user model, that provides a deep understanding or rich picture (from Soft Systems Methods) of the management consultancy process and embodies the user's requirements. It focused on management consultants working individually, in a non-prescriptive manner across a range of domains. It aimed to identify what consultants do in their natural setting, how they do it, and what and how outside influences (contextual factors) affect the process. It involved the following:

- A review of the management consultancy literature.
- An in-depth case study of a specific management consultant, using documentary records kept by the consultant, semi-structured interviews with the consultant, and observation of role-plays with the consultant.

Two broad research strategies may be considered (Weitzman and Miles, 1995): the case oriented approach focuses on a narrow, in-depth understanding of a high number of variables for a low number of (often single) cases; whereas the variable oriented approach focuses on a broad, shallow understanding of a low number of (often one or two) variables for a high number of cases (as shown in figure 2.7).

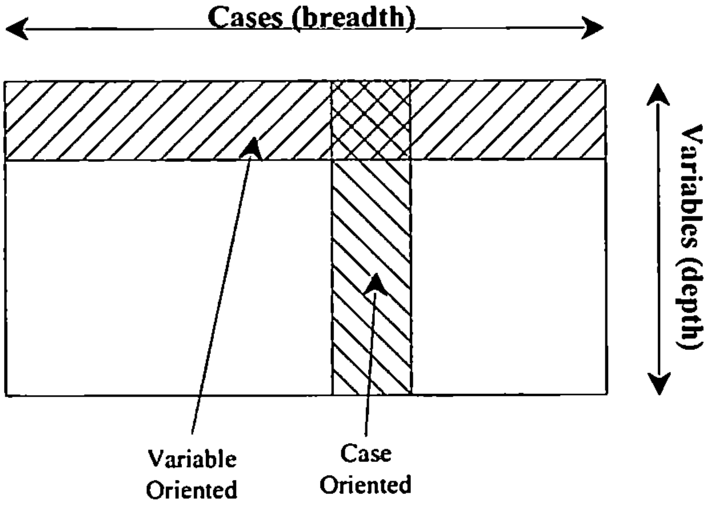


Figure 2.7 - Case Oriented and Variable Oriented Research Strategies.

Yin (1994) describes the case study strategy as being most advantageous for research that seeks in-depth explanatory understanding of complex contemporary phenomena within their 'real life' context, where the investigator has little control over events. This matches the above mentioned aim of the present work. He also identifies three main criticisms of the case based approach: that it provides a poor basis for scientific generalisation, it involves a long drawn out process resulting in massive unreadable documents, and previous work has suffered from a lack of rigour, typified by sloppy methods with conclusions influenced by equivocal evidence or biased views.

The present work addressed these issues by:

- employing a triangulation approach, that is to say using data from different sources to test and validate the developing user model; that it was accurate, and gave both a deep and broad understanding of the management consultancy process.
- including full transcripts of observations as appendices rather than including them in the main text.

- ensuring that the vast majority of statements can be grounded in specific paragraphs within these transcripts, which are numbered and cross referenced in the text.

The user model, which emerged from these stages, was represented using natural language narratives, a data flow diagram (DFD), and a user-centred rich picture diagram.

2.2.1.1 Literature Review of Management Consultancy

A literature review of the field of management consultancy was conducted in a holistic manner; that is to say it examined a wide range of issues across the consultancy field rather than narrowly focusing narrowly on an area that was anticipated as being amenable to computer support. This was done in order to identify potentially important contextual information regarding the consultancy process and determine the generality of aspects identified by the case study (whether they were specific to the individual consultant, common to subsets of the consultancy field, or common to the consultancy field as a whole).

2.2.1.2 Interviews with the Consultant

Discursive semi-structured interviews were used during the early stages of information gathering to provide background information and an overview of the assignment process.

2.2.1.3 Documentary Records of Consultancy Assignments

Very little documentation from management consultancy assignments was available; a significant amount of information is retained in the minds of individual consultants, or is recorded as informal hand written notes, which may be difficult to locate and interpret. Even when documentation does exist, consultants are often reluctant to submit it for analysis for reasons of sensitivity and client confidentiality.

2.2.1.4 Observation of Role-play Exercises

The objectives of these exercises were to elicit a complete set of sample data relating to a manual consultancy assignment (background, interview notes, and feedback material), and information relating to details of two key typical activities of the management consultancy process (information gathering via an interview and information analysis done back at the consultant's offices). The confidentiality of consultancy assignments prevented direct observation of real activities in naturalistic setting; thus, a role play approach was taken.

The consultancy interview and the analysis session were chosen as the activities to be role-played because of their significance in the consultancy assignment process and their relatively low resource requirement. Although other roles-plays (such as feedback meetings with members of the client organisation) might also provide insight into the consultancy process they would require many more people. This would be far more difficult to organise, monitor, and analyse.

During these role-play exercises the consultant was asked to vocalise his thoughts. This is similar to the 'thinking aloud' technique, which has been used for usability evaluation (described in section 2.1.3.3).

2.2.2 Identification of Disciplines Related to Management Consultancy

This activity (Described in more detail in chapter 4) aimed to identify other disciplines that may contribute to the development of CAMC software. This involved the following:

- The identification of analysis methods similar to those used by this type of management consultant, but originating from other disciplines, particularly the human sciences.
- A critical review of the usefulness to this type of management consultant, of the functionality provided by existing software designed for the above analysis methods.

Both of these were done primarily via literature review. However, an interview with the consultant was conducted to identify and confirm which analysis methods were similar to those used by this type of management consultant, and to discuss the suitability of the software support that existed in these disciplines. Also, two particular qualitative data analysis software tools were empirically examined. The software review was not done by empirical usability evaluations with the management consultant, because this would have taken too much of the user's time. There were too many software tools for a single user to evaluate: it is likely that the user would become confused. In addition, it could have biased the user's evaluation of the prototype software tool, by giving him a pre-conceived idea of how software should look and behave. Also, as this activity was at times running in parallel with activities seeking to elicit the consultant's manual methods, prolonged use of software tools could have started to change his methods. Lastly, extensive use of software by the consultant could have pushed the interviews toward discussion of technical issues rather than overall usefulness.

2.2.3 Synthesis of a Generic Design Model for CAMC software tools

This activity (described in more detail in chapter 5) sought to develop a generic model of CAMC software tools that included a generic software requirements specification (described in section 2.1.1), and a generic design rationale, which would 'be critical to' those 'who want to build upon the system and the ideas it embodies' (Carroll, 1997: p. 72). It involved the following:

- A literature review of CAMC.
- Development of a prototype CAMC software tool.
- Development of a software requirements specification for the prototype CAMC software tool.

- Formative iterative design evaluations.

2.2.3.1 Literature Review of CAMC

A literature review of CAMC software tools was conducted in order to identify issues that may contribute to the development of a generic model of CAMC, such as types of software tools; effective and ineffective design strategies; appropriate and inappropriate computing technology; and benefits, limitations, and impact of such software tools on the consultancy process.

2.2.3.2 Prototype CAMC Software Tool

An executable prototype CAMC software tool was developed in order to clarify the generic model of CAMC (acting as an executable software requirements specification), and act as a vehicle for empirical evaluation of the model by the user. This was done as the consultant interacting empirically with a working software tool (including user interface and functionality) was considered essential for eliciting a realistic evaluation of usefulness, usability, and impact on the consultancy process.

Formal specifications were not used as they can be difficult for users to understand, and cannot be used to represent the user interface (Vonk, 1990). Paper based models were not used for the same reason.

2.2.3.3 Software Requirements Specification for Prototype CAMC Software Tool

This was done in parallel with the development and evaluation of the prototype, and is included as Appendix I. It acted as a written record of the facilities that the software tool provided and as such was under continuous revision in parallel with the prototype. It provided a summary of the software's functionality.

2.2.3.4 Formative Iterative Design Evaluations

The aim of this stage was to identify and eliminating usability and learnability issues relating to the prototype software tool. However, to a lesser degree clarification of usefulness relating to the general model of CAMC software support also occurred.

This stage was done as part of the iterative design (described in section 2.1.3.3) process: modifications were made to the prototype as a result of comments that were elicited during repeated evaluations; effectively 'exercising the prototype' (described in section 2.1.3.2). Thus, it was formative (described in section 2.1.3.3): done in order to improve the prototype.

Two types of evaluation (described in section 2.1.3.3) were used: heuristic evaluations by technologists and empirical user testing by the management consultant. The heuristic evaluations began almost as soon as the prototype was started, with the user testing being introduced in later iterations, when many of the more technical problems had been resolved.

Heuristic Evaluation by Technologists

Many evaluations were conducted, using three evaluators (with experience in both academia and industry): a human factors specialist and two software engineers. Explicit guidelines (usability heuristics) were not used in order to preserve the varied perspectives of different evaluators.

Empirical User Testing by Management Consultant

Four evaluations were conducted with a single test user. This was done in order to identify issues that only emerge out of empirical use with a real user. The same experienced management consultant who was involved in the previous exercises was used because of his computer literacy, previous experience in evaluating software tools, and familiarity with present work. In ideal circumstances a larger number of users would be involved, being representative of all user types: however, this was not possible due to the time required to repeat the test with many other users and no other consultants being available.

2.2.4 User Acceptability Evaluation

This activity (described in more detail in chapter 6) aimed to evaluate the general model for CAMC software tools via the consultant's empirical use of the prototype CAMC software tool. This stage was summative (described in section 2.1.3.3): done in order to summarise the impact of CAMC software tools by management consultants. Of primary concern was the usefulness of CAMC software tools to management consultants. It sought to draw a comparison between the manual and software assisted processes, considering the impact of the software on the consultancy process. Care was taken to focus on issues relating to the underlying model of user requirements rather than issues relating specifically to the prototype. It involved a empirical user test followed by an acceptability interview.

2.2.4.1 Empirical User Test: Software Assisted Analysis Exercise

A user test (described in section 2.1.3.3) was undertaken involving 'simplified thinking out loud' (described in section 2.1.3.3) to allow comparison with the manual analysis exercise, and provide the user with practical experience of a CAMC software tool as a basis for the following acceptability interview. The user test was an empirical laboratory based exercise.

2.2.4.2 Acceptability Interview with Management Consultant

An in-depth interview was conducted with the management consultant after the software assisted analysis exercise. Its aim was to evaluate the general model for CAMC software tools, by eliciting information regarding the usefulness to management consultants, limitations, and changes to the consultancy process that may result from the use of CAMC software tools. Hence, the elicitation of subjective user satisfaction was focal.

2.3 Summary

This chapter argues that a combination of new and old methods should be used to develop software for creative application domains, such as management consultancy. It presents an overview of the research methods used, and the rationale for their use. The overall approach taken was user-centred, participative, and holistic.

3. Management Consultancy: Towards a User Model

This chapter presents a user model of the management consultancy process, which describes the main features of the characteristics and practices of management consultants. It considers aspects such as the tasks undertaken, methods employed, and difficulties encountered. It begins by providing an account of the research method used in the study. It then presents a review of the management consultancy literature, followed by a description of the results of a case study of a particular management consultant. It concludes by drawing this information together to give a descriptive user-centred rich picture of the field of management consultancy.

3.1 Research Method

The activity involved a review of management consultancy literature, and an in-depth case study of a specific management consultant (hereon referred to as the consultant). The case study used semi-structured interviews, 'thinking aloud' role-play observations, and documentary records kept by the consultant.

It was undertaken as a systems analysis study, which used methods drawn from the social sciences, it was not undertaken as a social science study: the objective was to elicit information regarding user behaviour that would be used to develop useful software; there was no intention to consider the sociological aspects of the user's practices in general.

3.1.1 Literature Review of Management Consultancy Field

A literature review of the management consultancy field was conducted at the same time as the interviews and role-play observations. This was done to provide additional contextual information regarding the consultancy process and to determine the generality of the data collected during the interviews and observation of role-play exercises.

3.1.2 Interviews with the Consultant

The work started and continued with many informal, discursive interviews with the consultant, undertaken over a three-year period. Three, more formal, semi-structured interviews were later conducted, during which audio recordings were made. These were transcribed verbatim, but are not included in this thesis for reasons of confidentiality (they refer to details of specific assignments and client organisations). The first of these interviews lasted two hours, the next lasted three hours and the final interview lasted two hours. These were undertaken over a nine-month period.

3.1.3 Documentary Records of Consultancy Assignments

During the first of these interviews, the consultant was asked to supply samples of documentary records relating to consultancy assignments. Samples of hand-written consultant's interview notes were supplied. However, it was not possible to use them directly, as it was difficult to interpret the handwriting, and to understand the meaning of the notes (being composed of heavily abbreviated ungrammatical sentences). A typed feedback sheet was also obtained (which is included as Appendix E).

3.1.4 Observation of Role-play Exercises

The objectives of these exercises were to elicit a complete set of sample data relating to a consultancy assignment (background, interview notes, and feedback material) undertaken manually (without software support), and information relating to details of two key typical activities of the management consultancy process (information gathering via an interview and information analysis done back at the consultant's offices).

Two interview role-plays were conducted, with the consultant playing the role of the interviewee, and an experienced systems analyst playing the role of the interviewer. The consultant was asked to provide sample background material relating to a fictitious client organisation, based on a amalgamation of previous assignments (included as Appendix B).

The systems analyst posed questions from an interview plan (included as Appendix A), which was draw up by the consultant as being representative of the topics/questions he would typically cover. The consultant responded to these questions based on his experience of responses given to similar questions by interviewees during previous assignments. In this way, the responses may be regarded as being representative of a range of previous real life consultancy assignments. This allowed live data from previous assignments to be collected without betraying client confidentiality. The role-play was observed by the researcher, and an audio recording was made.

The recording of this interview was then played back to the consultant in real time (several weeks later - enough time for his memory of his responses to fade - simulating, to a certain degree, coming to the interview fresh). As the consultant listened to the tape, he made interview notes (as he would in a real interview). He was also encouraged to vocalise his thought process, in particular to describe what he would record, and how he would have conducted the interview. An audio recording was made of this session, which included all the material from the interviews with the consultant's analytical annotations inserted at corresponding positions. This tape was then transcribed verbatim (included as Appendix C). A set of consultant's interview notes (included as Appendix D) were then made up from the actual interview notes recorded by the consultant and the consultant's oral descriptions of what else he would have recorded, which appear on the transcript.

Following this, an analysis session role play exercise was conducted. The consultant was asked to analyse the interview notes generated by the role play interview exercises, with a view to present his findings during a (fictitious) workshop session with the client organisation's management team. As with the previous exercise, he was asked to 'think aloud'. The role play was observed by the researcher, and a video recording was made of the session, which was then transcribed verbatim (included as Appendix E).

3.2 Literature Review

Management consultancy can be defined as the provision of an independent advisory service to a client, by an individual or group (referred to from here on as consultant(s)), in the field of management.

3.2.1 The Client

The term client can have both an institutional and a personal meaning. It can refer to the organisation that purchases the services of a consultant (also referred to as the client organisation) and to the individual(s) who initiate those services (Tisdall, 1982; and Kubr, 1986).

Consulting services are provided to private sector organisations, such as banks, manufacturing companies, and retail companies; public sector organisations, such as local and central government agencies; and social organisations, such as hospitals, universities, schools, religious organisations, charities, and trade unions (Hyman, 1961; Tisdall, 1982; Kubr, 1986; Blake and Mouton, 1990; and Rassam and Oates, 1991).

3.2.2 The Consultant

3.2.2.1 Organisational Change

Consultants are closely associated with change in organisations (Argyris, 1970; Kubr, 1986; Blake and Mouton, 1990; and Clark and Salaman, 1998): Tisdall (1982) describes change as a fundamental reason for the existence of consultants. An assignment will usually be prompted by a situation which is judged to be unsatisfactory by the client, and ideally end with a change taking place that is seen by the client to be an improvement (Tisdall, 1982; Kubr, 1986; Schein, 1988; and Margerison, 1988).

3.2.2.2 What they provide

When consultants are called into a client organisation they may provide the following:

Experience, knowledge, and skill

Hyman (1961), Margerison (1988), Rassam and Oates (1991), and Markham (1994) describe consultants as having knowledge and skills that can be dispensed to a client organisation. Consultants will pass through many organisations and, as a result, encounter general trends within organisations and common causes of problems (Hyman, 1961; and Kubr, 1986). This allows them to bring a body of experience and knowledge to the client organisation, which has been 'distilled from a portfolio of similar cases' (Tisdall, 1982: p. 96).

Management systems, methods and technologies are constantly being added to and updated. Consultants are closely associated with the diffusion of new management systems, methods, and techniques coming from universities and research institutions (Tisdall, 1982; Kubr 1986; and Rassam and Oates, 1991). Examples include the following: quality circles, management by objectives, job enrichment, work study, matrix structures, diversification of business, restriction to core business, 'just in time' stock control, and supply chain management (Tisdall, 1982; and Margerison, 1988; and Rassam and Oates, 1991).

Tisdall (1982), Kubr (1986), Margerison (1988), Schein, 1988, and Markham (1994) describe essential skills that consultants should possess, which have both technical and human components. The technical component includes analytical and problem solving skills: the ability of the consultant to analyse large complex problems and match them to management systems, methods and techniques in a short time is critical to the success of an assignment. The human component includes interpersonal, communication, and diplomacy skills: it is essential that the consultant is able to listen to the client, identify body language and verbal cues, and articulate findings and recommendations effectively in a sensitive environment. The importance of creative thinking and imagination is also discussed.

An impartial outside view

Hyman (1961), Tisdall (1982), Kubr (1986), Schein (1988), Blake and Mouton (1990), Rassam and Oates (1991), Markham (1994), and Czerniawaska (1999) describe the independence and objectivity of consultants, which has a significant impact on the consultancy process. It allows them to act as an impartial arbitrator or referee. Proposals coming from consultants may be acceptable because they come from an unbiased outside source. It can also allow them to identify options that had not occurred to members of the client organisation because they see the situation from a different perspective. It can also give consultants access to information from members of the client organisation that they would not supply to others within the organisation, especially management: hence management's perspective of an organisation is often very different to the actual situation. The consultant therefore has the ability to gather information for a client organisation's management team that would otherwise be unavailable.

Temporary executive staff

Consultants can also be called in to provide additional managerial staff on a temporary basis (Tisdall, 1982; Kubr, 1986; and Rassam and Oates, 1991).

3.2.2.3 Independent Advisory Service

The consultants' responsibility is for the quality and integrity of the advice they give. They have no direct responsibility or authority for deciding if or which changes are implemented: the decision to act on the consultant's advice has to lie with the client (Hyman, 1961; Argyris, 1970; Tisdall, 1982; Kubr, 1986; Margerison, 1988; Schein, 1988; Clark and Salaman, 1998; and Czerniawaska, 1999). The consultant should not be in a position where their ability to give honest, objective advice is restricted (Tisdall, 1982; and Kubr, 1986), even if, for example, one of the recommendations is that the client organisation's founder and chief executive should step down. Tisdall (1982) describes an example of this, involving Tesco and the McKinsey consultancy practice.

3.2.2.4 Internal and External Consultants

Consultants may be internal (also referred to as permanent or in house consultants), where they exist within an organisation to provide consulting services to other units of the same organisation, or external, where they exist within an organisation that provides consulting services to other organisations (Tisdall, 1982; Kubr, 1986; and Margerison, 1988).

3.2.3 The Client-Consultant Relationship

Kubr (1986) describes the recognition of the importance of the client-consultant relationship as 'a golden rule of consulting'. Tisdall (1982), Kubr (1986), Margerison (1988), and Clark and Salaman (1998) describe the critical importance of the client-consultant relationship, as without it the consultant's advice may not be understood or believed, and hence implemented by the client. This could deter the client from requesting the consultant's advice in the future, which is extremely significant to the consultant as it is estimated that around 70% of consultancy work is repeat business from established clients. The client-consultant relationship is based heavily on trust.

The consultant must be trusted by the management and members of the client organisation for an assignment to succeed. This trust stems from the client's confidence in the consultants' technical ability and integrity (guarantee of confidentiality). This integrity operates at two levels:

- At an organisational level the consultant will not make public (especially to competitors) sensitive information, which assignments inevitably involve (Hyman, 1961; Markham, 1997; and Rassam and Oates, 1991).

- At a personal level the consultant will often keep the sources of information confidential from the client (Markham, 1997). Blake and Mouton (1983) describe the consultant 'being careful not to reveal the identity of particular interviewees' during presentation (feedback) meetings, because 'participants are usually prepared to be frank when their identities are concealed'.

As a result of this, a certain level of secrecy surrounds consultants and their work, which can sometimes make it difficult to give examples of their work (Hyman, 1961; Tisdall, 1982; Rassam and Oates, 1991; and Czerniawaska, 1999).

3.2.4 Consultancy Styles

There are many classifications of consultancy styles, for example Blake and Mouton (1983) distinguishes between five consultancy styles. However, for the purposes of the current work it was useful to distinguish between two fundamental styles: prescriptive and participative.

Although there has been a strong shift toward the participative style (Tisdall, 1982; Kubr, 1986; and Margerison, 1988), it is recognised that it is necessary to use a style that is appropriate to the client's situation, and that a prescriptive style is still appropriate in certain circumstances (Blake and Mouton, 1983; and Kubr, 1986).

3.2.4.1 Prescriptive Style

Tisdall (1982: p. 118) refers to this style of consultancy as the 'technique push approach', Kubr (1986: p. 44) as 'resource' consultancy, and Margerison (1988: p. 27) as 'arms length' consulting. Participation by the client is typically focussed on content rather than process: discussing progress, and accepting or declining proposals (Kubr, 1986; and Margerison, 1988). The consultant is left to do the work on behalf of the client – to produce the answer (Kubr, 1986; and Margerison, 1988). It typically results in the production of a written report (Margerison, 1988; and Rassam and Oates, 1991).

An advantage of this style is the relatively small amount of time the client has to give to the consultancy assignment (Margerison, 1988). A common problem associated with this style of consultancy is that the client does not understand or 'own' the consultant's recommendations, and hence does not implement them (Tisdall, 1982; Margerison, 1988; and Rassam and Oates, 1991).

3.2.4.2 Participative Style

This style of consultancy is referred to by Kubr (1986: p. 44) as 'process' consultancy, and by Margerison (1988: p. 27) as 'arm in arm' consulting. The focus is on participation of the client in the assignment: the client and consultant working on the problem together, and producing joint analysis and proposals (Tisdall, 1982; Margerison, 1988; and Schein, 1988; and Rassam and Oates, 1991). The consultant helps the client to solve their own problems by passing on his approach, methods, and values (Kubr, 1986; and Schein, 1988).

3.2.5 Consultancy Assignments and the Consultancy Process

The term assignment (or project) is used to refer to a piece of self contained work undertaken by a consultant on behalf of a client. Tisdall (1982: p. 111) comments that 'there is no such thing as a typical assignment' as they can vary quite considerably in their 'size and complexity' (Tisdall, 1982: p. 106). Hyman (1961: p. 5) indicates that they can range from 'very brief and simple difficulties' to 'comprehensive reorganisations lasting several years'.

3.2.5.1 Levels of Intervention

Markham (1994) describes four levels of intervention for consultancy assignments: Purposes, Issues, Solutions, and Implementation. These levels of intervention relate to how well defined the assignment is: This may range from the client feeling that something isn't right, where the consultant helps determine where problems may exist, to the client knowing what is wrong and which technique or method needs to be applied to improve the situation, where the consultant provides specialist expertise regarding the technique or method concerned.

Purpose

Tisdall (1982), and Markham (1994) describe a level of intervention consisting of the clarification and development of the assignment's purpose (aims, or goals).

Identification of Issues

Tisdall (1982), Blake and Mouton (1983), Kubr (1986), Margerison (1988), Markham (1994) describe the identification (definition or diagnosis) of major issues (opportunities or problems) within the client organisation.

Recommendation of Solution

Another common level of intervention for consultants is producing recommendations (proposals or action plan) to address the major issues (Hyman, 1961; Kubr, 1986; Margerison, 1988; and Markham, 1994). Margerison (1988) comments that the best recommendations are frequently a mixture of ideas from the client and consultant.

Implementation

Although it is less common, the consultant may be asked to assist with the implementation of the recommendations (Kubr, 1986; and Margerison, 1988).

3.2.5.2 Phases, Stages, Activities, and Steps

There are several models of the stages (or phases) in a typical assignment: Kubr (1986) describes a five phase model, Margerison (1988) presents a model consisting of twelve steps grouped into three stages, and Markham (1994) treats his four levels of intervention as phases in consultancy assignments and describes a model with three activities. They show a high degree of variation in the overall way assignments are conducted, in terms of the content, focus, and sequence of stages. Although all models present the stages as a sequence, several report a high degree of overlap between stages (Kubr, 1986; and Margerison, 1988).

For the purposes of the current work is was useful to group activities into two broad categories: support activities and core activities. Support activities are those such as project management, marketing, sales, finance, briefings, discussions relating to the contract between the consultants and the client organisation, and reviews (evaluations) of the assignment.

The three core activities (shown if Figure 3.1) are those that are focal to the assignment, and are present (in one form or another) in all models of the consultancy process: information gathering activities, information analysis activities, and information presentation activities.

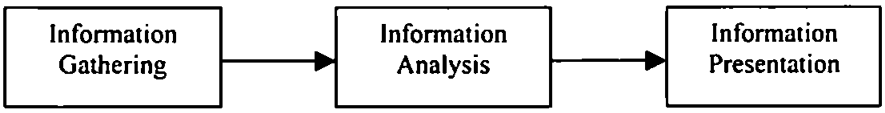


Figure 3.1 – Simple model of three core activities in management consultancy assignments.

The following sections describe these activities in more detail. However, there are relatively few detailed descriptions from practitioners of the actual processes of data gathering, analysis, and presentation.

Information Gathering Activities

Margerison (1988) describes a data collection step, Markham (1994) describes a data collection and analysis step, and Kubr (1986) describes fact finding as part of a diagnosis phase.

Information gathering in consultancy assignments can involve four fundamental types of activity: interviews, questionnaires, observation of client organisation, and review of documents, including organisational records and published material, (Argyris, 1970; Kubr, 1986; Markham, 1997; Schein, 1988; Margerison, 1988; and Markham, 1994). The face to face interview is one of the most frequently used information gathering methods during assignments (Margerison, 1988). Several authors (Markham, 1994; Markham, 1997; and Margerison, 1988) indicate the importance of eliciting information concerning both fact (hard data) and opinion (soft data). It would be very unusual for them to work with full verbatim transcripts of interviews or observations as this takes too long, and eliciting audio and video recordings interferes with the confidentiality of the process and hence the client-consultant relationship.

Information Analysis Activities

Margerison (1988) describes an analysis diagnosis step, Markham (1994) describes a data collection and analysis step, and Kubr (1986) describes fact analysis as part of a diagnosis phase. During this stage the information that has been gathered from several sources is collated, related, structured, and prioritised. Tisdall (1982) indicates that this activity is most often conducted back in the consultants' offices and not on the client's premises.

This stage involves the following tasks:

- Searching – Margerison (1988) describes the task of sifting through the data and identifying key words as part of the analysis process.
- Classification – Argyris (1970); Kubr (1986), Margerison (1988), and Markham (1994) describe creating a number of areas (groups, categories, or classes) into which the issues (themes or data) fall.

- Cross-checking – Kubr (1986) describes comparing (or verifying) information collected during one interview against information collected during another interview.
- Causal Analysis – Kubr (1986) describes causal analysis directly and Tisdall (1982: p. 110) describes 'identifying the source of the problem'.
- Charts – Several authors describe using visual techniques (diagrams and charts) to represent issues (Margerison, 1988), causal relationships between issues (Markham, 1994), and relationships between people and groups (Kubr, 1986; and Margerison, 1988).

Information Presentation Activities

Kubr (1986) describes presentation of proposals to the client as part of an action planning phase and Margerison (1988) describes a feedback step. During this stage the findings and/or proposals are presented (or fed back) to the client organisation. This can be achieved by either an oral presentation, or a written report, or both, and may involve discussion with the client (Kubr, 1986; and Margerison, 1988).

3.2.6 Difficulties Encountered by Consultants

Four of the most significant problems that consultants face are clients' failure to implement recommendations, information overload, time, and cost.

3.2.6.1 Understanding of Feedback and Failure to Implement by Client

The problem of the client not acting on the recommendations given by consultants is largely associated with the client not understanding the recommendations, or not owning them (Tisdall, 1982; Margerison, 1988; and Rassam and Oates, 1991). Essentially this is a problem of presentation; getting the client organisation's management team to understand what the consultant is presenting, to discuss it and thereafter to commit to it.

3.2.6.2 Inappropriate Application of Solutions by Client

Rassam and Oates (1991: p. 23) describe the problem of clients applying management concepts, techniques, and methods in inappropriate situations, which stems from them becoming 'fashionable' and being viewed by clients as 'panaceas for every management problem'.

3.2.6.3 Information Overload

There is an almost inexhaustible amount of information available in client organisations and consultants need to be selective with the information they collect to prevent it becoming unmanageable (Kubr, 1986; Margerison, 1988; and Markham, 1997). Markham (1997) also describes the need for consultants to be selective in what they include on illustrations that they present to the client: too much information can hinder rather than help understanding.

3.2.6.4 Time-scale

Data collection and analysis activities can take considerable amount of time to complete and consultants need to ensure that what they agree to deliver can be accomplished within the assignments time scale (Kubr, 1986; Margerison, 1988).

3.2.6.5 Cost

Kubr (1986) indicates that consultancy assignments can be costly, and this often deters organisations from hiring consultants, especially the smaller organisations (Tisdall, 1982).

3.3 Case Study

This section presents the results of the case study of a management consultant, which involved interviews with the consultant, observations of interview and analysis role play exercises, and review of the consultant's documentation. The main ideas that came out of the exercises are grouped according to which of the three core assignment activities (information gathering, analysis, and presentation) they relate to rather than which exercise they originated from. This is done for ease of reading. The source exercise may be identified via the paragraph numbers enclosed in square brackets.

3.3.1 Introduction

The subject of this case study initially worked in manufacturing industry as a project engineer. He spent six years in that post, followed by five years as a consultant with a large consultancy practice (employing about a thousand consultants). He then returned to industry (in the service sector) for seventeen years, initially as a works manager and then as a group engineering and planning manager, before moving to his current post.

At the time of the study he was a consultant working for a small external (described in section 3.2.2.4) consultancy company, following a participative style (as described in section 3.2.4.2). The practice had been operating for about ten years and employed five consultants. He had worked for them for about eight years.

He was selected as the subject of the case study on the basis of his willingness to be involved, total of thirteen years consultancy experience, twenty three years industrial experience, and the indication from the consultancy company for whom he worked that he followed typical management consultancy methods and practices.

The sample assignment is intervening at the level of identifying organisational issues (as described in section 3.2.5.1).

3.3.2 Assignment Stages

During interviews the consultant described the nature of assignments typical to the consultancy company he worked for. He indicated that the detailed structure of assignments varies considerably. However, the overall structure may be summarised by the following six stage model:

1. Preparation: Initial meeting (briefing/unstructured interview) between the consultant and the lead manager. During this meeting the scope of the assignment is defined (what issues are to be investigated, in which parts of the organisation), the interviewees are selected, and the client organisation's structure is identified (usually expressed as an organisation chart). Briefing of those involved in assignment. This is usually done on the client organisation's premises.

2. Information is gathered using semi-structured face-to-face interviews with members of the client organisation. One or two pilot interviews are usually conducted and the results used to modify the following interviews. This is usually done on the client organisation's premises.

3. The analysis of the information gathered and the preparation of material for feedback to the client organisation's management team. This is usually done back at the consultancy company's offices.

4. The feedback of information (issues and other information) to management team and those involved by consultant. At this point the results are fed back to the management team and the participants. This is usually done on the client organisation's premises.

5. A consensus regarding issues, which is reached through discussion between consultants and management team. The consultants then allow the management team to discuss the results in relation to the client organisation's business strategy statement. This provides the consultant with more information as the client uses their experience and background knowledge to analyse the results. A consensus is reached on what issues exist. The consultants record new information in the form of minutes. This is usually done on the client organisation's premises.

6. An action plan to tackle the issues identified above is developed and agreed by management team with consultant(s') advice. This is usually done on the client organisation's premises.

The stages are generally performed in the sequence shown, with some overlap and a significant amount of iteration [367]. Stages 4, 5, and 6 frequently occur during a single 'feedback' meeting or 'workshop' between the management team and the consultants. A typical time period for these activities would be between two and four weeks; during which time the consultant would 'live the data', i.e. he would be deeply immersed in it. He deliberately aims to keep the overall assignment time short and intense so that the assignment is completed before his memory starts to fade.

3.3.3 Information Gathering Activity

The scenario used for this activity was a one to one, face to face, questionnaire guided (semi-structured) interview, conducted on the client's premises by a single consultant working alone. The consultant indicated that this was the most common method employed by consultants, although observations of the client organisation and questionnaires were also used. Typically between ten and twenty members of the client organisation are selected to be interviewed. The consultant will attempt to get representations across levels and departments.

3.3.3.1 Identification of Issues

The consultant's overall aim is to elicit issues that are potentially important to the client organisation [30, 33, 48, 78, 86, 90, 92, 156, 158, 170, 178, 180, 202, 237, 242, 247, and 249]. These may be strengths, weaknesses (current problems), opportunities, or threats (potential problems). The consultant is aiming to elicit the interviewee's perceptions of themselves [15, and 106], other people, groups within the organisation (of which the interviewee may or may not be a member), and the organisation as a whole [29, 32, 55, 84, 89, 182, and 188], relationships between themselves and others, and relationships between others [37, and 178].

The consultant will usually identify and record issues (or potential issues) during the interview, based heavily on personal experience. The consultant commented on the importance of eliciting both facts and opinions from the interviewee in order to get a full picture of the client organisation.

3.3.3.2 Control

In general, the consultant controls the interview, in terms of the time it takes, and the information being given. This will sometimes involve moving the interview along [109, 115, 120], and getting the interviewee back on track [173, 159] if they diverge from the topic of interest. In general, consultants will allow an interviewee to wander more during earlier interviews [120], as this often yields valuable background information. If allowed to wander, later interviews tend to repeat material previously elicited; hence, the consultant tends to direct them more.

3.3.3.3 Questions

The consultant will ask the interviewee questions. Although the consultant determines the exact wording of many questions during the interview, for most of them the information to be gathered, or topic or subject area to be discussed is pre-determined and written down as a form to be filled in during the interview (a sample interview plan is included as Appendix A).

Supplementary Questions

Frequently additional (supplementary) questions are generated during the interview, in response to information given by the interviewee [18, 34, 41, 51, 56, 57, 64, 67, 70, 73, 78, and 217].

Some supplementary questions seek to confirm what the interviewee has said [3, 5, 205, and 210]. This often involves repeating the interviewee's comments back, or paraphrasing them [100]. The interviewee may change their mind, or revise a prior response to include more information or be more specific or put it more eloquently [60, 62, and 73].

Some supplementary questions seek to obtain additional information, usually more depth regarding a topic already mentioned or discussed in less detail [45]. Sometimes it will be necessary to elicit definitions of terms/vocabulary used by the interviewee [51], or details of a topic that initially seemed unimportant but which now seems relevant.

The consultant is occasionally required to answer questions posed by the interviewee [126]. This often occurs when an interviewee is unsure what was meant by a question posed by the consultant. Hence, the consultant sometimes has to provide an explanation for questions or terms used in a question, or to put it into terms that are more familiar to the interviewee. Where possible the consultant uses the vocabulary used by the interviewee [51].

3.3.3.4 Recording

As the consultant is listening to the interviewee speak, he analyses the responses, and records filtered/condensed/distilled notes of what they say [53, 60, and 61] consisting mainly of key words recorded as free text [61, and 108]; although pre-determined checklists are also used. It is often difficult to judge what is significant and what is not; as single statements often contain many ideas. The consultant does this using his prior knowledge and experience [69, 78, 107] in conjunction with the objectives of the assignment, which were given in the briefing [56].

He also sorts/collates what he hears. The interviewee will rarely give information in a coherent sequence, moving logically from one topic to another. Instead the information will be disjointed, a topic will be discussed, then another, then the original topic will be re-visited. In answering a given question interviewees frequently cover other topics [19]. The consultant will attempt to record each piece of information in the relevant place on the interview plan.

Notes in Margin

The consultant often records what he referred to as 'side notes' or 'margin notes' that consist of impressions of the state of the interviewee, such as defensiveness, frustration, evasiveness, and tension [75, 226, 229, 233, 242], or instructions to himself to take particular actions. These actions are usually cross-checking activities, such as 'check job description' or 'check this with John Smith', which indicate cross-checking to be conducted during other interviews, or against organisational documentation.

3.3.3.5 Relating Issues to People

The consultant also relates comments and issues to relationships between members of the client organisation, often via the organisation structure chart [30].

3.3.3.6 Cross Checking

He also cross-checks the different perspectives for coherency and divergence. He searches for other evidence: confirmatory or contradictory statements [25, 30, 45, 48, 56, 90, 93, 250].

3.3.3.7 Confidentiality and Trust

The consultant will also have to deal with any concern on the interviewee's part at being interviewed [86, 92]. This involves putting the interviewee at ease [82, 97], and establishing trust [154] through assurance of confidentiality.

3.3.4 Information Analysis Activity

The scenario used for this activity was a single consultant working back at base, analysing the results of the consultant's interview notes (elicited during the information gathering activity), and preparing material for presentation to the client (during the information presentation activity).

The consultant indicated that the process consisted of about 20% exploratory analysis and about 80% confirmatory analysis (described in Appendix H). He also commented that a considerable amount of time was spent collating interview notes, and selecting sub-sets of interviews and question responses.

3.3.4.1 Searching the Interview Notes

On several occasions the consultant can be seen to be searching through the interview notes [279, and 305]. Sometimes he was reading the interview notes in depth and others he was scanning them for key words and phrases. The consultant indicated that the interview notes functioned as an aide memoir, acting as a trigger causing him to 'recall a certain amount' from the interview [300].

3.3.4.2 Categorisation of Issues

The consultant described and spent a large portion of time 'listing' issues under 'headings' or 'categories' [272, 279, 294, 296, 313, and 348]. He used four main (or standard) headings: structure, job definition, communication, and relationships [272, 274, 287, 294, 320, and 350]. He occasionally created a 'sub-set of issues' within an issue [315], and indicated that if a particular issue seemed to be significant enough it could be included as an additional heading [289, 293, 294, and 302] providing a 'slice across' [293] or different perspective of the data. These additional headings and sub-issues were named using words from the consultants notes, which were often the actual words used by the interviewee.

He spent a considerable amount of time deciding how to structure the issues: deciding which heading to list a particular issue under, whether two similar comments were two separate issues or examples of the same issue, whether an issue was a sub-issue of another or an issue in its own right or an opposing view [272, 279, 281, 285, 350, 358, and 388]. He also commented that this 'takes a lot of time to draw ... out' [350].

3.3.4.3 Cross Checking

The consultant frequently performed and discussed cross checking different interviewees comments relating to a common issue (to see whether they confirm or oppose each other), and each interviewee's comments for consistency [264, 265, 266, 272, 274, 279, 296, 298, 305, 313, 348, 350, 352, 367, and 379]. More specifically the consultant checks for reciprocal relationships: for example, does the reporting relationships indicated by an interviewee agree with the official organisation chart [264, and 266], or if an interviewee identifies another as a point of contact, has the other party described the reciprocal relationship during interview [296, and 298].

3.3.4.4 Weighting of Issues

The consultant described the importance of assigning weights to the issues. He described the significance of the number of times an interviewee mentioned an issue, and the number of times different people raised the same issue [274, 279, 281, 285, 305, 311, 350, 267, and 393].

3.3.4.5 Links between Issues, and Causal Analysis

The consultant performed and described identifying links between issues [274, 284, and 294]. One type of link between issues is that of causality [272, and 274].

3.3.4.6 Charts

The consultant described the production of two forms of chart as 'typical' [345-346]: the official organisation chart with other relationships overlaid, and a 'star chart' that he describes by saying 'you take the person you're considering as the nucleus and everybody else is on an orbit around' [338]. He comments that the star chart was developed to overcome a problem associated with overlaying relationships onto the official organisation chart: the limitation of the amount of information it can show without becoming 'visually complex' [334]. He indicated that the star chart is useful for showing the activity around 'hot spots' or 'key people' [336, and 342].

3.3.4.7 Selection of Presentation Material

The consultant indicated that in preparing material to present to the client organisation he would limit the number of items listed under each heading to 'no more than about 8 items or so' to avoid giving too much information at once [281, and 287], and he would be careful to present only what was important to the client organisation [285, 294, and 298].

3.3.5 Information Presentation Activity

The scenario used for this activity was a single consultant presenting feedback material to members of the client organisation on the client's premises over a two day workshop.

3.3.5.1 Accessibility

The consultant comments that during this activity 'what you're actually doing is trying to bring them on board with the issues as quickly as you can' [294], making the issues 'readily accessible to the management team' [313, 317, and 348]. He described the difficulty of trying 'to distil ... a lot of data ... without losing too much of the meaning' and that if the consultant presents too much information then 'people are just bemused by the amount of information you're' giving them [389]. He indicated that part of making the information accessible is having a 'clear understanding, certainly at MD level' of what will be discussed during the activity [315], which may involve fitting the issues into the client's 'perception of' the situation, 'how they want to tackle it' [317, and 320]. He comments that 'if you start off with something' different to 'what is expected ... then ... bringing people on board' will be far more difficult [320].

3.3.5.2 Stimulating the debate: Cross checking

A significant aspect of the feedback meeting is described by the consultant as 'stimulating the debate' [269, 281, 294, and 298]. This involves cross checking the data elicited during the interviews: identifying whether some of the issues are known [294, and 298], and whether they are accepted practice or something that needs 'addressing' [267, 285].

3.3.5.3 Confidentiality and Trust

The consultant mentioned the confidentiality of the interviewee several times [274, 281, 294-298, 352-353, 356, 358, 360, 362, and 364]. He commented that the feedback would 'be fairly carefully worded' in order to protect the source(s) of information [294], and the information would be fed back as coming from 'two or three different quarters' without mentioning who they were [352]. He indicated that occasionally during interview he would ask 'Do you mind if that's made public ... that it was you that actually said it?' [296], and that it was normal to reveal the source of straightforward questions such as 'Who do you informally report to?' [298].

3.3.6 Accuracy of the Exercises

When asked how close to the real situation he felt the exercises were, the consultant replied by saying that 'the first interview in particular was probably fairly close to a real situation', and 'the second interview we curtailed a little bit' [383].

He commented that the main limitation of the exercises was that they only involved two interviews [271, 313, 350, and 383], which made the analysis look 'extremely simple' whereas with ten or twenty interviews it was more complex and took 'a lot of time to draw' the issues out and structure them [350]. He also indicated that with more interviews the cross checking would have stood out more, and the numeric weightings would be more realistic (because very few issues were common to both of the interviewees as they were in different parts of the organisation) [350].

3.4 Conclusions

This section draws together and reviews the data from the literature review and the case study, which was presented in the previous sections. This provides evidence that places management consultancy on the left hand creative side of the application domain continuum (described in section 1.1); specifically, this includes the importance of knowledge, experience, and skills (such as personal judgement, creativity, and intuition) in the consultancy process; and the highly individualistic and hence variable nature of the management consultancy process (summarised in section 3.4.2).

3.4.1 Overview

The literature review and case study, presented in the previous section, were used to develop a model of consultancy practice, which is summarised by the user-centred rich picture diagram presented as Figure 3.2. It represents the consultant's mental model of the management consultancy process (elicited via the case study) enhanced, clarified, and put into context by the literature.

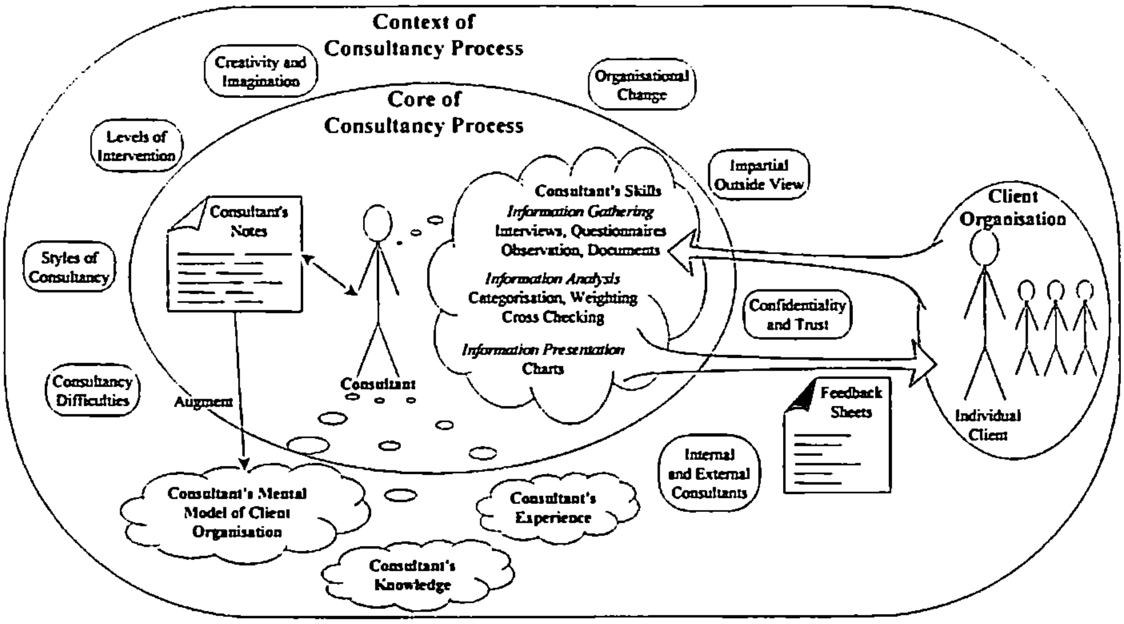


Figure 3.2 – Model of the Management Consultancy Process: a User Centred Rich Picture Diagram.

At the centre of the diagram is the consultant (the user): the diagram is quite literally user-centred. The core of the consultancy process focuses on data stores and sub-processes. It consists of the consultant, the tacit tasks that the consultant performs during an assignment (involving essential consultancy skills, such as communication, interpersonal, and analytical skills), and a set of interview notes, which act as an *aide-mémoire* triggering memories of relevant information from previous interviews and meetings with the client. In this way the interview notes augment the consultant's mental model of the client organisation.

The context of the consultancy process surrounds the core of the consultancy process. This context consists of several factors that have a significant influence on the core. These factors include: the style adopted by the consultant, the level of intervention for the assignment, difficulties experienced by the consultant, the consultant's creativity and imagination, and the confidentiality and trust between the consultant and the client, on which the communication of information (via documents such as feedback sheets) depends. The following sections describe several aspects of this model in more detail.

3.4.2 Experience, Knowledge, and Skills: Underlying Core Sub-processes

The literature (section 3.2.2.2) indicated that three key aspects of management consultancy are the things that the consultant brings to the client organisation (a model of which is shown in Figure 3.3).

Management Consultancy		
Management Knowledge Methods & Techniques	Experience of Many Client Organisations	Skills: Information Gathering Information Analysis Information Presentation

Figure 3.3 – Model of key aspects of Management Consultancy Process.

The literature (section 3.2.2.2) indicated that management theory, methods and techniques are being constantly added to and updated.

The literature review (section 3.2.5) and the case study (section 3.3.2) indicated that the sequence of stages involved in a consultancy assignment is variable; not only across consultancy practices but also across consultancy assignments within the same practice. However, three core activities are common to most assignments: information gathering, information analysis, and information presentation.

Alternatively, three underlying concurrent sub-processes may be seen to be present across a wide range of consultancy practices, many consultancy assignments and most of the consultancy process: these three processes are represented in Figure 3.4 as a data flow diagram. It shows information from members of the client organisation being recorded as interview notes, subsequently analysed and the results of this analysis fed back to the client organisation. It is important to understand that these processes occur across different activities; during an information gathering activity the consultant will be required to listen to the interviewee (information gathering sub-process), understand what he or she is saying (information analysis sub-process), and summarise what has been said in order to confirm understanding (information presentation sub-process).

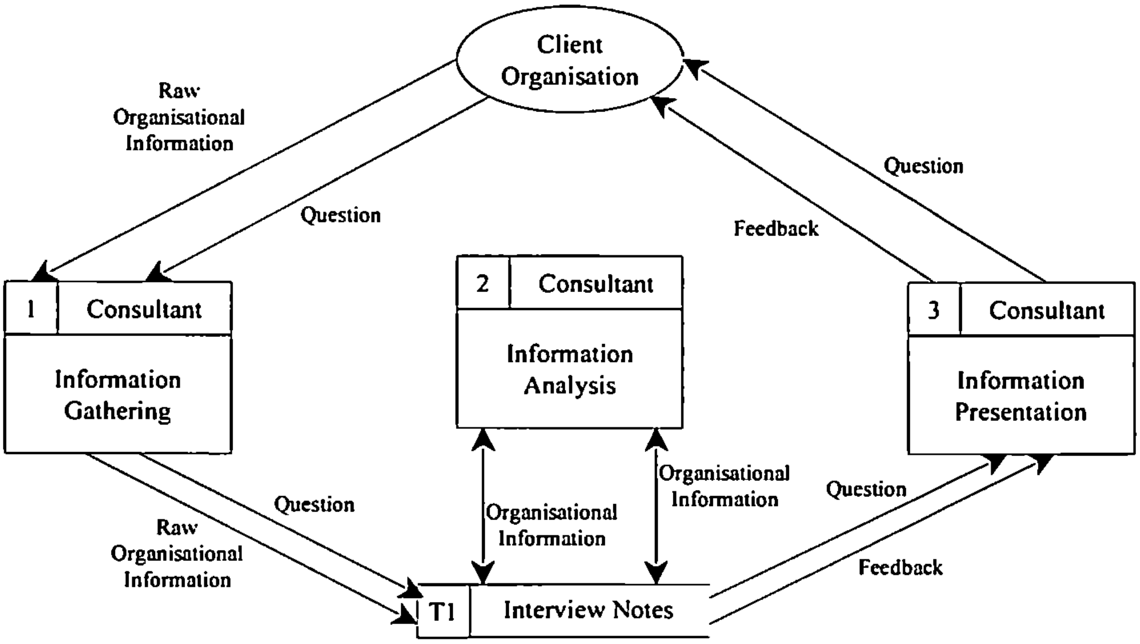


Figure 3.4 – Data Flow Diagram of the consultancy process.

The information gathering sub-process involves the consultant listening to what the client says during interviews, questionnaires, observations, meetings, telephone conversations, and reading company documentation.

The information analysis sub-process involves the consultant filtering, structuring, collating, cross checking, and relating all of the organisational information gathered from multiple sources by information gathering process.

The information presentation sub-process involves the consultant presenting organisational information back to the client: both by speaking to them (during interviews, meetings, etc.), and writing to them (via letters, memos, or reports).

The consultant's interview notes represent a model of the client organisation, which augments the consultant's mental model of the client organisation.

3.4.3 Integrated Quantitative and Qualitative Analysis

The consultant perceives the organisational information as a set of inter-related information, not as distinctly separate sets of quantitative and qualitative data: the literature (section 3.2.5.2) and the case study (section 3.3.3.4) describe the organisational information as including responses selected from checklists, responses relating to relationships between people and groups, free text responses and free text annotations.

The analysis of such information oscillates rapidly between qualitative, quantitative, and integrated perspectives: this can be seen in both the literature (described in sections 3.2.5.2) and the case study (described in sections 3.3.4), and includes the following activities:

- Collating and filtering interview notes according to interviewee, plan, and question
- Searching the interview notes for concepts and phrases. In particular those relating to organisational issues, key words, people and groups.
- Categorising of responses in the form of free text in multiple ways, thus relating sections of interview notes to organisational issues.
- Cross-checking sections of the interview notes relating to or connected with an issue in some way, for consistency within an individual's interview, and for agreement/contradiction across several interviewees.
- Generating graphs representing the relations between people, and groups, indicated by the interview notes.
- Cross checking for reciprocal relationships.
- Quantitative weighting of qualitative issues, which calculates the number of times each issue was mentioned, and the number of people commenting.
- Categorise/classify of organisational issues hierarchically.

- Relate organisational issues to one another, such as on the basis of a causal relationship.
- Relate organisational issues to people and groups in the client organisation.
- Relate organisational issues to the graphs

3.4.4 Confidentiality

The significance of confidentiality was described in the literature review (section 3.2.3) and the case study (section 3.3.3.7). It exists at two levels: the organisational level, where the consultant agrees not to reveal information relating to the organisation to competitors, and the personal level, where the consultant agrees not to reveal information that individuals have given during for example interviews, to others in the client organisation.

3.4.5 Difficulties Encountered by Consultants

The literature review (section 3.2.6) and the case study (sections 3.3.3.4, 3.3.4.6, and 3.3.5.1) described several problems that the consultant faces during an assignment. One of the most significant is information overload – the information gathering activities (such as interviews) can yield far more information than it is possible to analyse and present. Another key problem relates to effectively communicating findings and recommendations to the client organisation's management team within a very short time period. Lastly, the cost of consultancy services was discussed, which is often beyond the budgets of smaller organisations.

3.4.6 Accuracy and Generality of User Model

A single consultant has been used. The responses given by this consultant are likely to be representative of every assignment in which he has been involved, and all other consultants whom he has come into contact with (either by working directly with them, discussing issues at seminars and conferences and read about). He is not somebody who has worked on a single assignment and had no contact with other consultants. As described in section 3.3.1, he was selected partially on the basis of his many years experience working in management and in both large and small management consultancy practices. He is likely to have come into contact with a large number and broad range of management consultants.

The concurrence between information yielded by the case study relating to the practices and characteristics of the specific consultant and the information regarding the variance and norms with the consultancy field by the management consultancy literature indicate that the specific consultant was a typical example of an external consultant working in a participative and non-prescriptive style; examples of this include the use of interviews as the most frequently used information gathering technique, and the reliance on repeat business.

3.5 Effectiveness of the Research Methods

This section considers the effectiveness of the research methods used to analyse and represent the characteristics of management consultants (described in section 3.1).

The literature regarding the field of management consultancy provided useful contextual information regarding what consultants do and why they do it (such as the importance of client confidentiality in the management consultancy process). This facilitated the development of a holistic perspective of consultancy practices. Also, it provided a broad overview of the consultancy field, which gave a clear indication of the generality of the information from the case study of the specific consultant; it helped identify which information was particular to the specific consultant studied and which was applicable to a wider cross-section of management consultants. This helped place the specific consultant within the consultancy field as a whole. However, it contained few details of exactly how consultants work; the specifics of the consultancy process.

The interviews gave a more focused picture of the overall workings of the specific consultant, such as the methods and techniques he used and the typical structure of his assignments. However, the consultant found it difficult to articulate the details of how he worked. Part of the difficulty being due to his concern not to reveal any of his clients' confidential information, which made it extremely difficult for him to illustrate comments with concrete examples. However, the main problem of articulation was that this is tacit knowledge deeply embedded in the consultant's practices, based on experience, intuition, imagination, and creativity. Much of what he was saying would have probably made sense to other consultants, but not to software developers.

Some of the documentation (the feedback sheets and charts) provided concrete examples of the deliverables from part of the consultancy process. The charts gave clear indication of how they were produced by the consultant; as it involved a fairly mechanical task that related to the interview questions in a straight forward manner. However, the feedback sheets gave no details and few indications of the processes and tasks involved in their production: how the consultant created them. Although examples of hand written consultant's interview notes were provided they were of limited use as it was difficult to read and interpret the hand writing.

The role play observations were used in response to being unable to observe the consultant working with his clients in the natural setting or in the laboratory for reasons of confidentiality. In order to overcome this the consultant (user) was asked to play the role of the other person. This proved very effective, opening up the specifics of the consultancy process and thereby revealing important tacit knowledge that was not elicited via the interviews or literature review. For example they revealed that weights are used to identify how localised to an individual or group the issues are, rather than reliability (which is not really appropriate to consultants as all views are reliable, but may not be held by everyone).

There is a layering effect with what was elicited by the different information gathering methods. The literature review gave a broad picture of the consultancy field, which included contextual information that significantly influences how consultants operate at lower levels. The interviews with a specific consultant gave information that was more focused on their overall characteristics, practices, and methods (for example it identified them as external consultants, and the literature gave meaning to this label). The documentation provided details of what the specific consultants produced at the end of the consultancy process. The interview and analysis role play exercises opened up the detail of key sub-processes in the consultancy process and revealed how they were related to the context; they provided concrete examples that gave meaning to comments appearing in the literature and occurring during the interviews. It was mainly by these means that the true nature of the consultant's internal analytic processes, and their contribution to his mental model of the client organisation became apparent.

The DFD was useful during the design and initial implementation of the prototype CAMC software tool (described in chapter 5) for module and database design, due to its focus on the core software development concepts (processes and data stores). However, it was limited in the range of aspects of the consultancy process that it could represent.

The User Centred Rich Picture was able to describe the consultancy process far more fully. In particular it was able to describe and highlight the contextual aspects of the consultancy process; such as the consultant's mental model of the client organisation, and the role of confidentiality in the consultant's interaction with the client. However, generating the user-centred rich picture was more difficult than generating the DFD: there were more options for what could be represented, the process of selecting what to represent was therefore more complex.

The high level of user participation in this work contributed to the amount of detail elicited regarding subtle tacit processes. This enabled the systems analyst to see the process from the consultant's point of view by extensive and detailed exposure to an example assignment, with concurrent explanation of the consultant's rationale; thus facilitating the articulation of what is normally tacit knowledge. The combination of methods employed here enabled the acquisition and refinement of a more accurate and complete model of the users' practice, which placed management consultancy on the creative left hand side of the application domain continuum (described in section 1.1).

3.6 Summary

This chapter presents a user model of management consultancy that describes characteristics and practices of management consultants, which identifies that it is based on human creativity, experience, intuition, and integrated quantitative and qualitative analysis. This provides empirical case study and literature based evidence that management consultancy is an example of a creative application domain.

4. Disciplines Related to Management Consultancy: Methods, Techniques, and Software Support

This chapter describes investigations of disciplines that are related to the field of management consultancy. This was to see if features of software tools developed for these disciplines could be of use to management consultants in addressing problems identified in chapter 3 and thus contribute to an enhanced user requirements specification for CAMC software tools presented in chapter 5.

4.1 Research Method

This activity involved literature reviews, an interview with the consultant that discussed potential usefulness of qualitative data analysis (QDA) and social network analysis (SNA) software tools (referred to as the technology interview), and empirical reviews of qualitative data analysis software. Two packages in particular (ATLAS/ti and NUDIST) were empirically examined in detail.

4.2 Organisational Behaviour

The field of organisational behaviour (OB) is concerned with the study of human behaviour in organisations (Greenberg and Baron, 1997; and Rollinson *et al*, 1998). It is driven by both academic and industrial interests; being concerned with the elicitation of knowledge regarding the behaviour of people in organisations as an end in itself and as a means to improve the competitiveness and effectiveness of organisations by solving organisational problems. Cole (1995) and Rollinson *et al* (1998) specifically mention the involvement of consultants from outside the organisation as change agents who guide and facilitate the change process.

Many theories, methods, and techniques have been developed aimed at improving organisational effectiveness (Greenberg and Baron, 1997; and Rollinson *et al*, 1998), such as management by objectives (MBO), quality circles (QC), and total quality management (TQM). These theories, methods, and techniques are under constant refinement in response to the changing nature of work; caused by factors such as changes in the workforce and current technology.

The field of OB draws heavily from the concepts, theories, methods, and techniques of the social and behavioural science (Greenberg and Baron, 1997; and Rollinson *et al*, 1998). It uses the following data collection methods: interviews, questionnaires, observation, and examination of documents; and analysis techniques. It uses both quantitative and qualitative research methods (including naturalistic observation and case studies, which are described in section 4.4.1.1). Rollinson *et al* (1998) specifically mentions the contribution of ethnography (described in section 4.4.1.1) in eliciting a rich and detailed picture of events thus allowing insight into issues of causality and how people make sense of their work.

4.3 Social Network Analysis

4.3.1 Introduction

Wasserman and Faust (1994) and Berg (1998) describe the field of social network analysis (SNA), also referred to as sociometry, which is concerned with concepts and methods that support the analysis of interaction between social entities.

These social entities (referred to as actors) may be individual people, groups of people, organisations (government bodies, charities, or private business), or nations. The collection of actors involved in a study is referred to as the actor set. Most studies use an actor set consisting of actors that are of the same type (for example, all people in group, or all groups within an organisation).

Actors are linked together by relational ties. These relational ties may be formal relations, biological relationships, physical connections, transfers of material (such as imports and exports between nations) or transfers of information (such as communication of order details between organisations). A relation is defined as a collection of relational ties of a specific kind. For any actor set several different relations may be measured.

A social network consists of finite set(s) of actors and the relation(s) between them. Social network data can include two types of variable: structural and compositional. Structural variables measure relational ties of a specific kind between pairs of actors (such as friendships between school children, or trade between nations). Compositional variables (also known as actor attributes) measure characteristics of actors (such as a person's gender, geological location, qualifications, or date of birth; or a nation's gross national product, population density, or land mass). Social network data should include at least one structural variable.

Relations (structural variables) may be directional or non-directional; and dichotomous or valued. With a directional relation each relational tie between a pair of actors has an origin and a destination: for example, it is possible for one child to consider another child to be their friend, but for the other child not to consider them to be a friend. With a non-directional relation each relational tie between a pair of actors does not have a direction (or may be perceived as existing in both directions): for example if two children live close to each other. Dichotomous relations are either present or absent for each pair of actors, whereas valued relations can take on a range of values indicating the strength, intensity, or frequency of each relational tie between pairs of actors.

An ego-centred network (also referred to as a personal network or local network) is focused on a single actor (referred to as ego) and includes other actors (termed alters) who have relational ties to ego. For example, a set of actors may each be asked to identify the people with whom they have recently discussed matters of importance. The set of responses from each actor would form a personal network.

4.3.1.1 Notation for Social Network Data

Graph Theoretic

In this notation each relation is represented by a graph of nodes connected by lines. A graph consists of a set of nodes (N) that represent the actors, and a set of lines or arcs (L) that represent the ties between actors. Arcs are directional connections, and are represented by an ordered pair of actors, such as <Bob, Jane>. Lines are non-directional connections, and are represented by an unordered pair of actors, such as (Sarah, Alan). The position of nodes on a graph is arbitrary. Network data consisting of more than one relation may be represented by several graphs. Each of these graphs may be shown separately or combined to form a single graph that uses different types of lines to represent different relational ties. An example of a network described using this notation follows:

The set of actors (N) = {Alan, Susan, Alex, Rod, Sally, Jon}

Relation 1 (Friendship) = {<Rod, Susan>, <Alan, Susan>, <Susan, Alan>,
<Susan, Jon>, <Jon, Sally>, <Alex, Sally>}

Relation 2 (Lives Near) = {(Rod, Alan), (Alex, Susan), (Alex, Sally)}

Sociometric

In this notation each relation is represented by a two dimensional matrix (referred to as a sociomatrix) where the rows and columns refer to the actors. This notation may be viewed as complementary to the graph theoretic notation, as sociomatrices are adjacency matrices for sociograms: each entry in the sociomatrix indicates whether or not two nodes would be adjacent (connected) on a sociogram. It can easily represent valued relations; the entries in the sociomatrix can easily take on non-dichotomous values. Network data consisting of more than one relation may be represented by several sociomatrices, which may be combined to form a single three dimensional matrix, referred to as a super-sociomatrix. It is difficult to represent actor attributes using this notation. This notation is by far the most common in the social network literature.

Algebraic

In this notation each relation is represented by a distinct capital letter, and a tie between two actors on a relation is represented by *iFj*, where *F* is the relation, and *i* and *j* are actors. This may be viewed as a shorthand for graph theoretic and sociometric notation. It is used to study multiple relations, and cannot handle valued relations or actor attributes. It is useful for using algebraic techniques to compare and contrast measured relations. The focus of such algebraic techniques is on the associations among the relations measured on pairs of actors, across the entire set of actors.

An example of a network described using this notation follows, where F represents the relation regarding friendship, and L represents the relation regarding location:

$_{\text{Rod}}F_{\text{Susan}}$

$_{\text{Susan}}F_{\text{Jon}}$

$_{\text{Jon}}F_{\text{Sally}}$

$_{\text{Alex}}F_{\text{Jon}}$

$_{\text{Susan}}F_{\text{Alan}}$

$_{\text{Alan}}F_{\text{Susan}}$

$_{\text{Rod}}L_{\text{Alan}}$

$_{\text{Alex}}L_{\text{Susan}}$

$_{\text{Alex}}L_{\text{Sally}}$

4.3.2 Information Gathering

Wasserman and Faust (1994) and Berg (1998) describe the variety of methods use in SNA to gather information. These include interviews, questionnaires, observations, and archival records. The questionnaire is the most frequently used method.

4.3.3 Information Analysis

There are a large number of mathematically (often statistically) based methods for analysing social network data (Wasserman and Faust, 1994; and Berg, 1998). The focus of social network analysis is on patterns within relations. There are methods that relate to actors themselves, for example how prominent an actor is within a group (as quantified by measures such as centrality and prestige). There are methods applicable to pairs of actors and the relational ties between them, referred to as a dyad.

A dyad may be described as being in one of three states. A null dyad has no arcs of a given type between the two nodes (no relational ties exist between the two actors). With an asymmetric dyad an arc of a given type exists in one direction or the other only (a single relational tie exists between the two actors in a specific direction). In a mutual or reciprocal dyad two arcs of a given type exist one going in one direction and another going in the opposite direction (two relational ties, going in opposite directions, exist between the actors).

4.3.4 Information Presentation

Wellman and Berkowitz (1988), Wasserman and Faust (1994), and Berg (1998) describe two main techniques for representing social network data visually: sociograms and sociomatrices.

4.3.4.1 Sociogram

The sociogram is a commonly used method for presenting social network data visually, and is based on graph theoretic notation (described in section 4.3.1.1). Actors are represented as nodes (points in a two dimensional space) and relational ties between actors are represented by arcs (lines between the nodes). The location of points on the page is arbitrary. Figure 4.1 gives an example of a sociogram showing relational ties between actors (the solid arcs represent the friendship relation and the dotted arcs represent the lives near relationship).

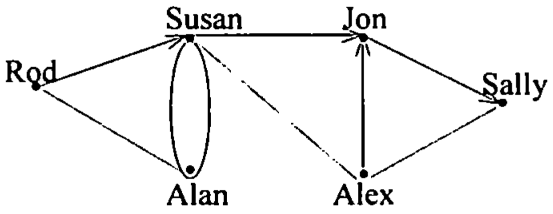


Figure 4.1 – Sociogram showing relational ties between actors.

4.3.4.2 Sociomatrix

Another frequently used method of presenting social network data is the sociomatrix, which is based on sociometric notation (described 4.3.1.1). Table 1 shows an example of a network described using this notation

-	Alan	Susan	Alex	Rod	Sally	Jon
Alan	-	1	0	0	0	0
Susan	1	-	0	0	0	1
Alex	0	0	-	0	1	0
Rod	0	1	0	-	0	0
Sally	0	0	0	0	-	0
Jon	0	0	0	0	1	-

Table 1 – A sociomatrix for relation 1 (friendship).

4.3.5 Computer Aided Social Network Analysis

Wasserman and Faust (1994) describe seven software tools designed for social network analysis. Most focus on providing facilities for performing calculations (such as centrality) on network data.

4.3.5.1 KrackPlot

KrackPlot (Krackhardt, Lundberg, and O'Rourke, 1993) provides facilities for creating, modifying, storing, and printing directed sociograms showing a single relation (described in section 4.3.4.1). Sociograms can be created either by adding nodes and lines to a blank sociogram, or by supplying KrackPlot with a textual input file consisting of node data and an adjacency matrix (described in section 4.3.1.1). It has the ability to place nodes in a circle around a central node, referred to as 'El Centro'.

4.4 Qualitative Data Analysis

4.4.1 Introduction

4.4.1.1 Styles of Qualitative Research

A significant portion of qualitative data analysis (QDA) literature is devoted to distinguishing and describing various qualitative research styles (also referred to as traditions or approaches), which are not necessarily mutually exclusive. Two such styles are ethnography and grounded theory (described by Tesch, 1990; Punch, 1998). Ethnography is concerned with describing and understanding cultures from an internal perspective (as viewed by the members of those cultures). It adopts a naturalistic approach where, as far as possible: the cultures are studied in their natural setting. Grounded theory may be viewed as a research strategy with supporting techniques that seeks to generate theory inductively from data. It does not try and fit data into a predetermined theoretical framework: instead the theory is developed from and hence grounded in the data.

This section does not consider styles of qualitative research in detail, instead it focuses more on styles and core characteristics of qualitative data analysis, which has direct relevance to the form of software that is likely to be most useful (Weitzman and Miles, 1995) and hence more relevance to the selection and design of software for management consultancy.

4.4.1.2 Characteristics of Qualitative Data and Analysis

Dey (1993), Miles and Huberman (1994), and Weitzman and Miles (1995) describe several important characteristics of qualitative data and data analysis, which influence the type of software support that is most useful.

Characteristics of Qualitative Data

A study may involve data consisting of a single case or multiple cases, being collected from a single source or multiple sources. Sources may be of different types: text, sound, still pictures, and moving pictures. The data may be fixed (such as official documents or historical records) or potentially subject to revision or correction (such as field notes). The data may be well structured (such as responses to a standard questionnaire or interview) or free-form (such as field notes).

Characteristics of Qualitative Data Analysis

The analysis may be exploratory in nature, where ideas evolve inductively, alternatively it may be confirmatory, where the study may seek to deductively test specific hypotheses derived from existing theory. The analysis may vary in how fine or coarse it is: it may use words, lines, sentences, and paragraphs as its data segments (described in section 4.4.3.3). Categories (described in section 4.4.3.4) may be pre-defined at the start of the study, perhaps based on theory, or may evolve as the study proceeds. The analysis may strictly apply a single category to a data segment, or data segments may have multiple categories assigned to them. The analysis may be comprised of a single pass through the data, or may involve an iterative style moving through the data several time taking different cuts. The analysis may also vary in the significance of the surrounding context of analysis units. Some qualitative researchers value 'closeness to the data' and deeply immerse themselves in it, while others doing more abstract work want to 'distance' themselves from the data (Weitzman and Miles, 1995: p. 14-15).

4.4.1.3 Conceptual and Mechanical Tasks

Jones (1985), Dey (1993), and Kelle (1995) distinguish between conceptual and mechanical tasks of qualitative data analysis. The conceptual tasks are essential to the process of qualitative data analysis being concerned with understanding the meaning of data. They are interpretative and creative in nature and hence difficult to describe explicitly. However, these tasks rely upon a large number of more mechanical tasks that are more easily expressed. These tasks tend to centre around the management of the large volume of interconnected data that is involved. They also represent a significant proportion of the time taken for analysis.

4.4.1.4 Problems

Two predominant problems associated with qualitative data analysis are described by several authors: information overload and decontextualisation (Dey, 1993; Miles and Huberman; and Kelle, 1995). The information gathering process and subsequent analysis usually generate large volumes of data. These need to be managed carefully in order to avoid becoming overwhelmed and confused. A key part of qualitative analysis is identifying and isolating key parts of the data. However, this process of removing data segments from their context can result in the data segments losing their meaning. This problem is known as decontextualisation.

4.4.2 Information Gathering

Several authors describe the methods and characteristics of collecting qualitative data (Dey, 1993; Silverman, 1993; Miles and Huberman, 1994; Silverman, 1997; Punch, 1998; and Silverman, 2000). Four main methods of qualitative data collection are identified: observation, interviews, questionnaires, and documentation. Interviews and observations may be recorded by making field notes or transcribing (frequently verbatim) from audio or video tapes. A key characteristic of qualitative research studies is the sustained time period of data collection (a year is not unusual).

4.4.3 Information Analysis

This section describes several tasks that are fundamental to qualitative data analysis. Although they are presented under distinct headings, in practice there is much iteration and overlap between them (Dey 1993; and Miles and Huberman, 1994).

4.4.3.1 Sorting and Filtering

Weitzman and Miles (1995) describe sorting data comprised of multiple cases and working with a sub set of the cases.

4.4.3.2 Annotations and Memos

Dey (1993), Weitzman and Miles (1995), and Kelle (1995) describe annotations and memos consisting of reflections and observations (frequently written in the margins of paper documents). For clarity this work will define annotations as being recorded during data collection, and memos as being recorded during subsequent analysis.

4.4.3.3 Data Segments

The data is broken down into units, referred to as data segments or chunks (Dey, 1993; and Miles and Huberman, 1994). Textual data may be divided into units consisting of words, phrases, lines, sentences, or paragraphs. A study may use one or many types of unit, and may allow units to overlap or be nested.

4.4.3.4 Categories

Many authors describe a fundamental task within qualitative data analysis (referred to as classification, categorisation, tagging, labelling, coding, and indexing) that organises and summarises the data, reducing it by identifying its essential themes, concepts, features, characteristics or qualities (Tesch, 1990; Dey, 1993; Miles and Huberman, 1994; and Strauss and Corbin, 1998).

Categories may be range between conceptual (such as bias or exaggeration) and empirical (such as a particular department, person, or product). An initial indication of potential categories and links with the text may be obtained by searching for key words or phrases in the source data. The categories used in a study may be pre-determined (from theory and/or the objectives of the study) or extracted (allowed to emerge) from the data during analysis, or a combination of both. A multiple (or inclusive) or single (or exclusive) category scheme may be used. Inclusive categories allow each data segment or case to be linked to multiple categories (for example a person's interests such as football, cricket, and rugby), whereas with exclusive categories each data segment or case may be linked to a single category only (for example hair colour such as black, brown, and blonde). Levels of subcategories may be developed giving more detail and hence more meaning to the analysis.

It is highly unlikely that a suitable set of categories and subcategories will be identified at the start of a study and remain unchanged. It is normal to follow a process of constant refinement, where categories are created, discarded and modified. Kelle (1995) describes generalisation as the process of combining two or more subcategories because it is not necessary to distinguish between them. This leads to greater integration and scope. He also