Performance Metrics for Network Intrusion Systems

By

CHRISTOPHER JOHN TUCKER

A thesis submitted to Plymouth University

in partial fulfilment for the degree of

DOCTOR OF PHILOSOPHY

School of Computing and Mathematics

Faculty of Science and Technology

April 2013

Performance Metrics for Network Intrusion Systems

Christopher John Tucker MSc

Intrusion systems have been the subject of considerable research during the past 33 years, since the original work of Anderson. Much has been published attempting to improve their performance using advanced data processing techniques including neural nets, statistical pattern recognition and genetic algorithms. Whilst some significant improvements have been achieved they are often the result of assumptions that are difficult to justify and comparing performance between different research groups is difficult. The thesis develops a new approach to defining performance focussed on comparing intrusion systems and technologies.

A new taxonomy is proposed in which the type of output and the data scale over which an intrusion system operates is used for classification. The inconsistencies and inadequacies of existing definitions of detection are examined and five new intrusion levels are proposed from analogy with other detection-based technologies. These levels are known as detection, recognition, identification, confirmation and prosecution, each representing an increase in the information output from, and functionality of, the intrusion system. These levels are contrasted over four physical data scales, from application/host through to enterprise networks, introducing and developing the concept of a footprint as a pictorial representation of the scope of an intrusion system. An intrusion is now defined as "an activity that leads to the violation of the security policy of a computer system". Five different intrusion technologies are illustrated using the footprint with current challenges also shown to stimulate further research. Integrity in the presence of mixed trust data streams at the highest intrusion level is identified as particularly challenging.

Two metrics new to intrusion systems are defined to quantify performance and further aid comparison. Sensitivity is introduced to define basic detectability of an attack in terms of a single parameter, rather than the usual four currently in use. Selectivity is used to describe the ability of an intrusion system to discriminate between attack types. These metrics are quantified experimentally for network intrusion using the DARPA 1999 dataset and SNORT. Only nine of the 58 attack types present were detected with sensitivities in excess of 12dB indicating that detection performance of the attack types present in this dataset remains a challenge. The measured selectivity was also poor indicting that only three of the attack types could be confidently distinguished. The highest value of selectivity was 3.52, significantly lower than the theoretical limit of 5.83 for the evaluated system. Options for improving selectivity and sensitivity through additional measurements are examined.

List of Contents

List of Figures

List of Tables

List of Equations

List of Symbols

Symbol	Definition
Ac	Accuracy
$D(\,)$	Distance in parameter space
FP	False Positive
FN	False Negative
$\mathrm{f(x|intrusion)}$	Probability density function conditioned on an intrusion present
$\mathrm{f(x|\overline{intrusion})}$	Probability density function conditioned on no intrusion present
i	Summation, product and vector element index
M	Number of intrusion event types
N	The number of dimensions in parameter space, number of signatures
$\mathrm{N_{Alert}}$	Number of frames alerted
$\mathrm{N_{Attack}}$	The number of frames in an attack
P	Probability
P_C	Probability of Confirmation
P_d	Probability of Detection
$\hat{P}_d$	Probability of Detection for an attack type
P_{fa}	Probability of False Alarm
$\hat{P}_{fa}$	Probability of False Alarm for an attack type
P_I	Probability of Identification
P_R	Probability of Recognition
Pr	Precision
$\mathrm{S_i}$	The set of frames that can violate the security policy of a network
$\mathrm{S_n}$	The set of frames consistent with the security policy of a network
$\mathrm{S_{sig}}$	The set of frames that will be detected by a given set of signatures
t	Decision threshold

TN	True Negative
TP	True Positive
U	The universe of frames present on a network segment
$X(\ )$	Probability vector for the triggering of a signature
x	The output variable from a NIS
η	Mean of probability density function
σ	Standard deviation of probability density function

List of Terms

Item	Definition
AAFID	Autonomous Agents for Intrusion Detection
Ac	Accuracy
AgNIS	Aggressive Network Intrusion System
AIDE	Advanced Intrusion Detection Environment
AMSEC	Attack Modelling and Security Evaluation Component
ANN	Artificial Neural Network
APT	Advanced Persistent Threat
ATC	Air Traffic Control
AVS	Anti-Virus Software
BERR	Department for Business, Enterprise and Regulatory Reform
BGP	Boundary Gateway Protocol
BSM	Basic Security Module
BST	British Summer Time
CAIDA	Cooperative Association for Internet Data Analysis
CFAR	Constant False Alarm Rate
CIDF	Common Intrusion Detection Framework
CMD	Command
COAST	Computer Operations, Audit and Security Technology
CPU	Central Processor Unit
CSV	Comma Separated Variable
DARPA	Defense Advanced Research Projects Agency
DDoS	Distributed Denial of Service

Item	Definition
DLC	Distributed Loosely Coupled
DNS	Domain Name Service
DoS	Denial of Service
D-S	Dempster Schafer
DSP	Digital Signal Processing
DTC	Distributed Tightly Coupled
DTI	Department of Trade and Industry
EST	Eastern Standard Time
FIN	Network packet type indicating completion of a connection
FIRE	Fuzzy Intrusion Recognition Engine
FN	False Negative
FP	False Positive
FTP	File Transfer Protocol
GA	Genetic Algorithm
GB	Gigabyte
GrIDS	Graph-Based Intrusion Detection System
HIDS	Host Intrusion Detection Systems
HMM	Hidden Markov Model
HTTP	Hyper-Text Transfer Protocol
IAP	Intrusion Alert Protocol
ICMP	Internet Control Message protocol
IDES	Intrusion Detection Expert System
IDMEF	Intrusion Detection Message Exchange Format
IDS	Intrusion Detection System

Item	Definition
IETF	Internet Engineering Task Force
IIDS	Intelligent Intrusion Detection System
ILC	Integrated Loosely Coupled
IP	Internet Protocol
IPS	Intrusion Prevention System
IPSec	Internet Protocol Security
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
IRC	Internet Relay Chat
ISBS	Information Security Breaches Survey
ITC	Integrated Tightly Coupled
KDD	Knowledge Discovery and Data Mining – A conference
KPMG	A global consultancy
LGP	Linear Genetic Program
MA	Mobile Agent
MAC	Media Access Control
MANETS	Mobile Ad hoc Networks
MARS	Multivariate Adaptive Regression Splines
MIB	Management Information Block
MIT	Massachusetts Institute of Technology
MLP	Multi-Layer Perceptron
MOAT	Measurement and Operation Analysis Team
MTU	Maximum Transmission Unit
NADIR	Network Anomaly Detection and Intrusion Reporter

Item	Definition
NATO	North Atlantic Treaty Organisation
NIC	Network Interface Card
NIDES	Next-generation Intrusion Detection Expert System
NIDS	Network Intrusion Detection System
NIS	Network Intrusion System
NIST	National Institute of Standards and Technology
NLANR	National Laboratory for Applied Network Research
NTP	Network Time Protocol
OS	Operating System
PBA	Polymorphic Blending Attack
P-BEST	Production-Based Expert System Toolset
PCA	Principal Components Analysis
PCI	Payment Card Industry
PCI-DSS	Payment Card Industry Data Security Standard
P_d	Probability of Detection
P_{fa}	Probability of False Alarm
PhD	Doctor of Philosophy
Pr	Precision
R2L	Remote to Local
RAID	Recent Advances in Intrusion Detection
RBF	Radial Basis Function
RFC	Request For Comment
ROC	Receiver Operating Characteristic
RIPE	Réseaux IP Européens

Item	Definition
RPC	Remote Procedure Call
RST	Reset – Network packet type
RSVM	Robust Support Vector Machine
SIEM	Security Information and Event Management
SMB	Server Management Block
SNMP	Simple Network Management Protocol
SOM	Self-Organising Map
SSE-CMM	System Security Engineering Capability Maturity Model
SSH	Secure Shell
SVM	Support Vector Machine
SYN	Synchronise – Network packet type
TB	Terabyte
TCP	Transport Control Protocol
TN	True Negative
TP	True Positive
U2R	User to Root
UDP	User Datagram Protocol
UK	United Kingdom
UTC	Coordinated Universal Time
VLAN	Virtual Local Area Network
VoIP	Voice Over Internet Protocol
VPN	Virtual Private Network
W3	What Where and When

Item	Definition
WAND	A network research group based in New Zealand
WITS	Waikato Internet Traffic Storage

Acknowledgements

The research described in this thesis is all the work of the author. Although nobody else has contributed to its content it could not have been completed without the support, encouragement and understanding of many others. First, I would like to acknowledge the invaluable support of my Director of Studies, Professor Steven Furnell. He continued to believe in the value of this research and my ability to complete it. His enthusiasm, professionalism and positive attitude have been inspirational.

Next, my supervisory team, Dr Bogdan Ghita and Dr Phil Brooke have provided that essential insight to the academic requirements necessary for completion of the research, guiding me towards successful submission of this thesis. Their contribution in proof reading the research papers and thesis drafts has been invaluable.

Finally, I would like to thank my family and friends for their encouragement over the seven years of study. In particular, without the tireless support of my wife throughout this period it is unlikely that it would have been finished. As we tackled the usual family challenges she never once complained at the time I spent, nor doubted that this thesis would be completed. In view of their unfailing support, this thesis is dedicated to my family; Yvonne, Rebecca, Edward and Rachel.

As I strived to complete this work I was inspired by an unexpected source. During my secondary education, at each morning assembly I faced a large wooden notice board on which our school motto was engraved as follows: "More men fail through lack of perseverance than through lack of ability". Good words indeed and essential advice for anyone contemplating part-time academic study.

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 without prior agreement of the Graduate Committee.

This study was privately financed, with the aid of Stochastic Systems Ltd.

A part-time programme of advanced research was undertaken, which included the theoretical assessment and practical measurement of metrics for network intrusion. Relevant conferences were attended at which work was presented; external institutions were visited for consultation purposes and several papers prepared for publication.

Word count of main body of thesis, excluding appendices: 42,545

Signed: .....

Date: .....

CHAPTER 1

INTRODUCTION

1. Introduction

At the outset of this research there was a perceived performance shortfall in practical systems, in particular that the number of false alarms was unacceptably high. Over thirty years of research into new data processing techniques and systems approaches has failed to identify a method or collection of techniques for correct declaration of all intrusions, with an arbitrary low false alarm rate. A key aspect of understanding the applicability of a new technique is the ability to compare it with the performance of others. The motivation for this thesis was based in the observation that detection theory, as a framework for performance assessment, had been applied successfully to many technologies, including radar and sonar, but had only limited application to computer intrusion systems. The desire was to use analogy from these other technologies to identify how performance could be better measured, specifically addressing the research question of how can performance of an intrusion system be defined more clearly to aid comparison of differing approaches and focus new research.

This research has defined a new taxonomy for intrusion systems, specifically aimed at comparing the performance of differing approaches. Five levels of intrusion performance have been defined from "Detection", which is the ability of a system to declare that an intrusion is underway but provide no further information, through to "Prosecution", where evidential quality information is gathered on the attacker and the methods and targets they have exploited. These five levels are examined over four data scales from application to

enterprise to create a footprint for each intrusion system type. Meaningful comparison of intrusion systems can only be undertaken where they overlap on this footprint.

In addition, this research has defined two new performance metrics, designated as sensitivity and selectivity, using analogy from other technologies in which detection is an important element. Sensitivity is concerned with the ability to detect a given attack, at a known detection probability and false alarm rate. It replaces the usual four metrics of false positive, false negative, true positive and true negative rates with a single number, making comparison easier. Selectivity is concerned with the ability of an intrusion system to differentiate different attack types. This is important when considering the higher levels of intrusion performance in the new taxonomy. Selectivity consists of a square matrix, the dimension of which is the number of different attack types that are to be differentiated.

This first section of this chapter commences by describing the need for effective intrusion systems in terms of the quantity, type and growth of network attacks. The aims and objectives of the research are then described more fully before an overview of the layout of the thesis is presented.

1.1. The Need for Effective Intrusion Systems

The growing availability of Internet access has increased the variety of online services offered to business and private users. The number of online financial transactions is continuing to increase to the point that Internet purchases and the management of personal finances online are commonplace. Businesses

have also identified significant benefits in interconnecting their company networks over the Internet or private networks, either to reduce infrastructure costs, improve employee-work flexibility or to collaborate with partners. However, as these opportunities have been identified and exploited, there has been an expansion of the number and sophistication of attacks on online users.

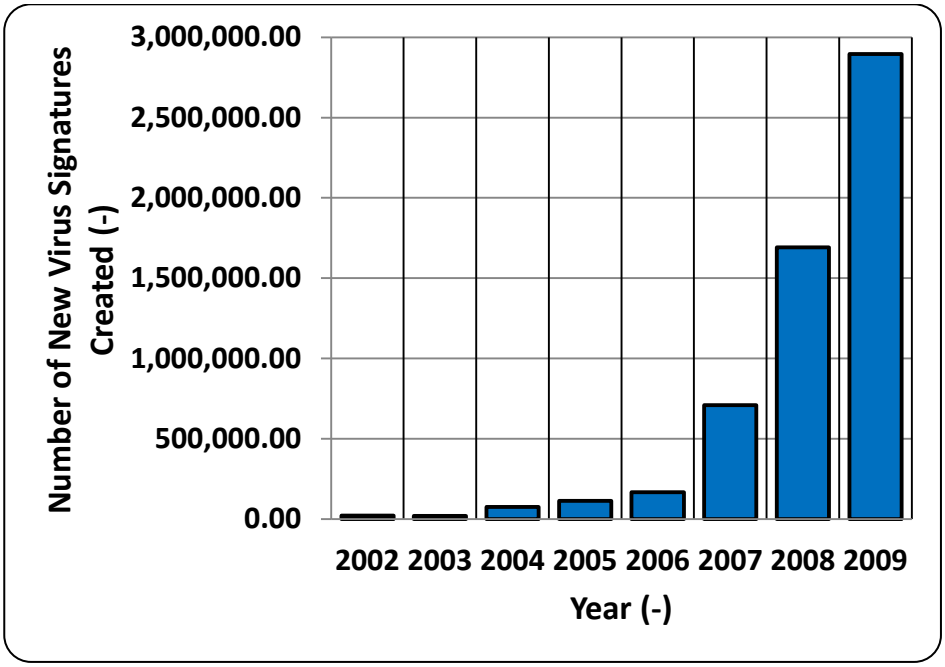


Figure 1-1 Annual Increase in the Number of New Virus Signatures

One indication of the growing threat is the number of new virus signatures created each year to combat malware. This is shown in Figure 1-1, as reported by Symantec (Rossi 2010), where the almost exponential annual increase in new virus signatures is shown. Despite publishing numerous statistics in their regular security reports, 2010 was the last time annual increases in virus signatures statistics were reported by Symantec. However, in a press release in 2011 (Symantec 2011), Symantec did report:

"The sheer volume of sophisticated attacks targeting organizations of all sizes poses a daunting challenge for traditional signature-based security solutions that can't keep up".

Also the number of new vulnerabilities discovered within security related software, as recorded in the US National Vulnerability Database (National Vulnerability Database 2011), is shown in Figure 1-2. The exponential growth in their discovery to 2006 seems to be abated, with a steady decline in the number since then. However in 2010 there were still over 4500 new vulnerabilities discovered, with each one presenting a potential attack vector for an intruder.

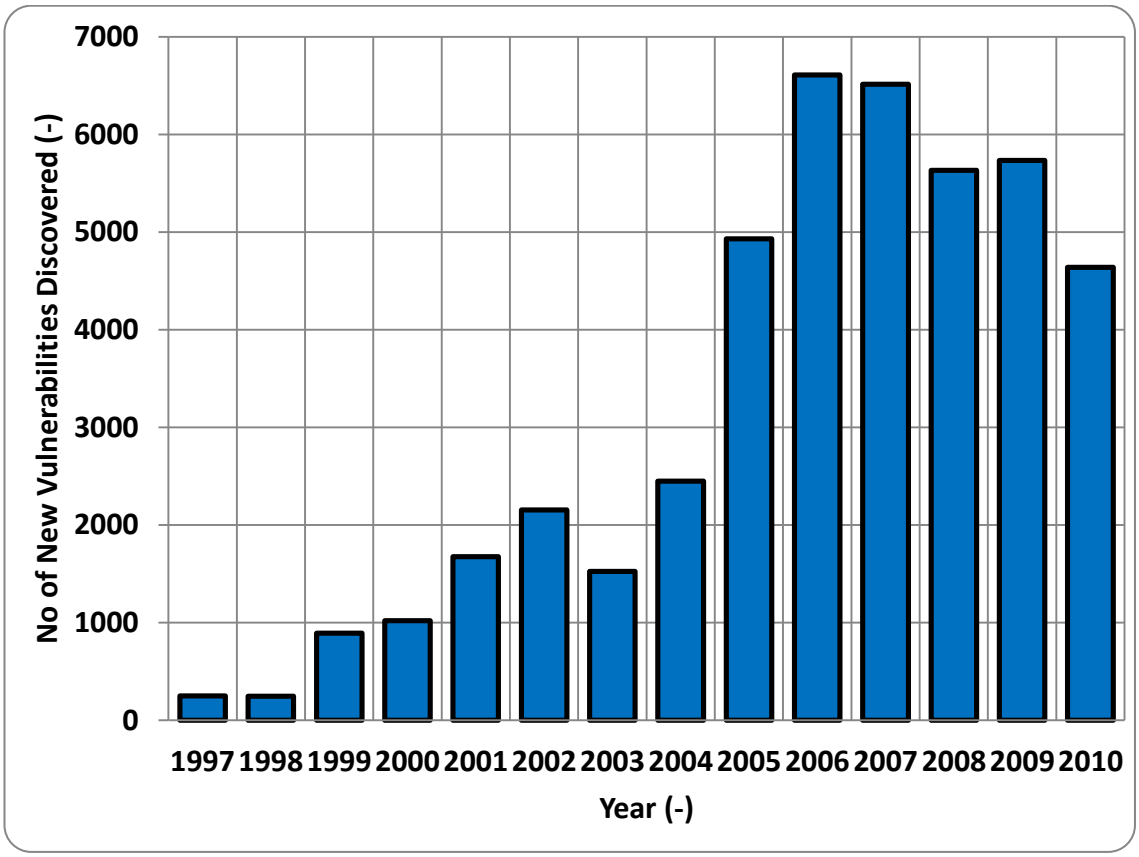


Figure 1-2 Annual Increase in New Vulnerabilities Discovered

The results of the latest UK bi-annual information security breaches survey are shown in Figure 1-3 (PricewaterhouseCoopers 2012). This survey shows that

the incidence of UK companies experiencing premeditated or malicious security incidents has increased from 18% overall in 1998 to 91% in 2012 for large companies. Interestingly, the 2006 incidents are lower than in 2004 where they reached 68% overall. The reduction is believed to be due to the increased awareness of security issues within UK companies and the deployment of improved security controls. However the reduction in incidents was abated in 2010, with 2012 representing the highest recorded level from this series of surveys.

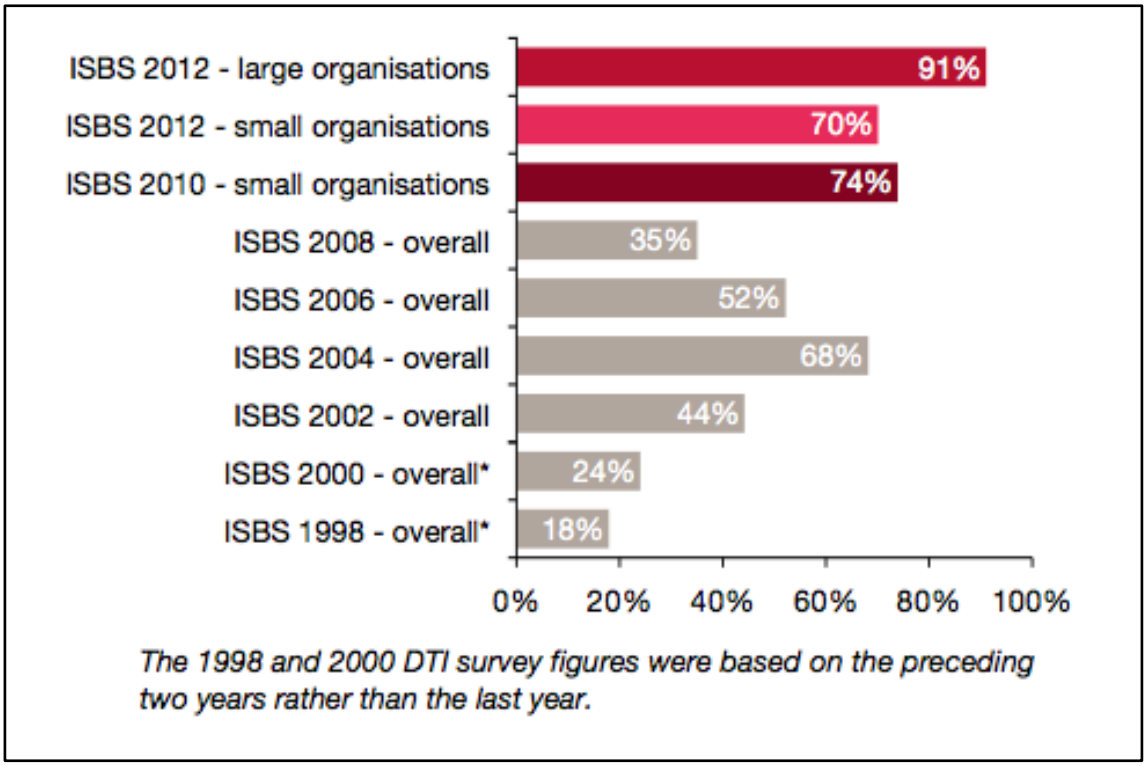


Figure 1-3 Results of a UK Security Survey (PricewaterhouseCoopers 2012)

Figure 1-4 shows a much more insidious problem in which the sophistication of attacks is increasing, but the technical skills required by would-be attackers is decreasing (Hansman and Hunt 2005). This is mainly due to the proliferation of scripts on the Internet that allow the automation of attacks. If these trends

continue, the importance of effective intrusion systems will also grow. Of particular relevance to the proposed research is the growth in the performance required from an intrusion system, to meet the growing threat and remain effective. Using Figure 1-4 as an example, an intrusion system in 1980 would have needed to detect password guessing attacks, whilst in 2000 it would need to detect sophisticated command and control attacks, as well as all the attack techniques developed since 1980.

More recently advanced persistent threat (APT) attacks have appeared, originally targeted at military and political targets, but now being used increasingly against large enterprises (Giura and Wei 2012). APTs are sophisticated attacks undertaken by highly skilled and resourced individuals, potentially attacking over many years for a given target.

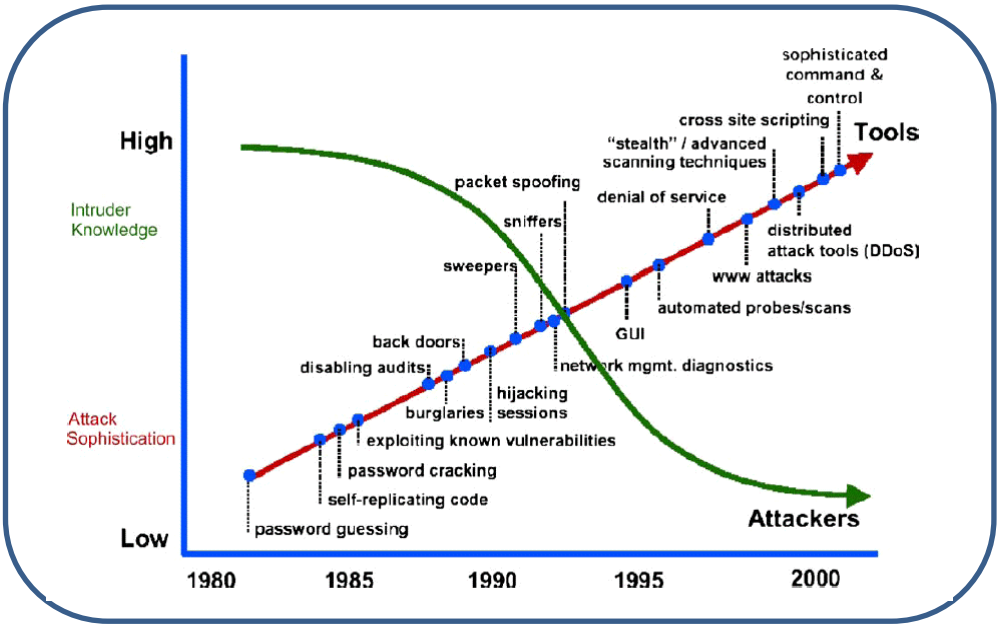


Figure 1-4 Increasing Attack Sophistication (Hansman and Hunt 2005)

As attack sophistication has increased so has the development of technologies aimed at defending network systems. Routers, which have formed the central

element of the Internet, have been enhanced to include packet filtering, stateful-inspection and authentication technologies. Specialist security devices have been developed, such as firewalls and intrusion systems and prevention systems, to thwart the would-be attacker. The deployment of such devices has made it increasingly difficult to penetrate protected networks, but not impossible. Configuration errors, technology weaknesses or security policy limitations can still result in networks being penetrated and information being stolen or otherwise compromised. In many cases legitimate system users access information that they are not authorised to see, either because no controls are in-place for "insiders" or through the use of simple hacking tools available on the Internet.

Clearly, the threat to computer networks is real, sophisticated and growing. New technologies may exacerbate the threat by providing opportunities for more configuration errors, new inherent weaknesses, or to reveal policy limitations. A key requirement in defending against intruder attacks is to know when they are underway. Intrusion systems are one way of providing this information, along with analysis of operating system, application and network device logs.

Despite this clear need for intrusion systems their deployment has not been universal. The 2008 BERR (PricewaterhouseCoopers 2008) survey[1] reported that only 46% of the UK companies surveyed had deployed an intrusion detection system, up from 43% in 2006. Yet 98% of companies use anti-virus

[1] This was the last survey in this series to record deployment statistics for intrusion detection systems.

software on email and web downloads, up from 95% in 2006. Given that there are quality open source intrusion systems available (for example SNORT or BRO), it is difficult to understand why a wider deployment of intrusion systems has not occurred.

One reason may be the perception that intrusion systems produce a large number of false alarms. An analysis by Tjhai (Tjhai, Papadaki et al. 2008) confirms that recent versions of SNORT (Roesch 1999) can still produce a large number of false positives. Also in the corporate environment Gartner (Young and Pescatore 2010) reports:

"False positive (false alarm) rates remain low in most deployments because most use cases deploy high confidence signatures only."

In the same study Gartner reports that 25% of intrusion prevention system (IPS) deployments have their blocking capabilities disabled until "businesses are confident that there will be no business interruption". This suggests that false alarm rates continue to remain a concern.

1.2. Aims and Objectives of the Thesis

The aim of this thesis is to identify better ways of measuring the performance of intrusion systems, so that meaningful comparison can be undertaken to focus future research on performance improvement.

In order to achieve this the programme of work wasdivided into four distinct objectives, namely to:

a) Review the techniques and performance measures that have been applied to intrusion systems, to identify the most promising techniques

for further evaluation;

b) Re-evaluate the meaning of "detection" in the context of Network Intrusion Systems (NIS[2]);

c) Assess the application of detection theory to NIS and propose metrics that can be used to characterise their performance; and

d) Demonstrate experimentally the use of the performance metrics and the potential for false alarm rejection using a representative NIS and practical data.

A programme of research has been undertaken to achieve these objectives and is reported fully in this thesis. A literature review was undertaken to determine the state of the art in intrusion systems, with emphasis on the detection of network intrusions, achieving the first objective of this research. The outcome from this review identified specific limitations of intrusion systems and their performance comparison, which were used to examine more closely the definition of "detection". From this research the new taxonomy was created and the need for specific performance metrics was identified, achieving the second research objective. These metrics were defined in a systems analysis based on detection theory, achieving the third research goal. The final research goal was achieved by designing and implementing a practical assessment in which these metrics were measured for the SNORT intrusion system on one specific dataset, namely DARPA 1999. Although this dataset is now somewhat dated it does

[2] The usual acronym of NIDS, Network Intrusion Detection System, has been replaced with NIS, Network Intrusion System, to indicate a more general applicability to other intrusion levels such as recognition and identification, as discussed in Chapter 3. In this thesis, detection is only used to describe the lowest level of intrusion, or when referencing the work of others which use it to describe their work.

demonstrate the application of these metrics and the resulting information that can be inferred. The proposed metrics are applicable to more modern intrusion threats and systems other than NIS.

1.3. Thesis Structure

The structure of this thesis mirrors the objectives of the research as defined in the previous section. Chapter 2 provides the literature review focusing on data processing techniques, as this is a focus of considerable interest in the research community. A short description of research and commercial systems is provided, before some security issues are described. The chapter ends with a description of the current methods for evaluating performance of intrusion systems. It is concluded that although there has been considerable data processing research it is difficult to compare intrusion systems directly in meaningful ways.

Chapter 3 uses the results of the literature review to develop in detail the new taxonomy that focusses on performance comparison of intrusion systems. It is not limited to network deployment, but covers scales of protection from files, to hosts, to networks and ultimately to the enterprise. The taxonomy also focuses on the use of the output from the intrusion system, defining five different levels of performance. When these levels are combined with the protection scale, a two-dimensional map is created, and the capabilities of different approaches to intrusion can be plotted. This represents a qualitative comparison of intrusion systems.

Chapter 4 describes a systems view of network intrusion using results based on detection theory. Some fundamental results are derived in terms of set theory, before the properties of an ideal NIS are defined. The two new performance metrics of sensitivity and selectivity are developed to quantify the ability of an NIS to detect intrusion events and to classify them by type of attack. This chapter builds on the qualitative comparison made possible by the taxonomy described in chapter 3, to provide a quantitative determination of performance.

Chapter 5 describes an experimental study that quantifies the performance of SNORT against the DARPA 1999 dataset, using the two new performance metrics. Poor performance was measured and used to indicate potential improvements to NIS processing.

Finally, Chapter 6 provides a summary of the research and conclusions, stating the contribution that this research has made to the corpus of intrusion systems. The potential for further work is also described.

CHAPTER 2

LITERATURE REVIEW

2. Literature Review

This chapter describes the current research in intrusion systems. During the last 33 years, since the original work of Anderson (Anderson 1980), there has been considerable research undertaken and therefore it is not possible to cover all aspects in such a short review. As a result this chapter concentrates on network intrusion research, relevant to the research objectives described in section 1.2. It is divided into five major sections, as follows:

- Frameworks for intrusion detection - in which basic types of intrusion systems are established;
- Data processing - describing the focus on algorithms that has occurred in trying to improve the discrimination performance;
- Intrusion systems - in which a systems level approach rather than a data processing centric view is undertaken;
- Intrusion evasion - describing the security problems introduced by deployment of an intrusion system and the techniques used by intruders to avoid detection of their activities; and
- Evaluating intrusion detection systems – in which the techniques and data available for quantifying intrusion system performance are described.

2.1. Frameworks for Intrusion Detection

The classification of different intrusion detection methodologies can be considered as frameworks from which their basic properties can be compared. There have been a number of attempts to define a generic architecture for

intrusion systems. DARPA initiated the Common Intrusion Detection Framework (CIDF) (Ning, Wang et al. 2000). In this framework, they are described in terms of event generators, event analysers, event databases and response units. The Common Intrusion Specification Language was created to operate with the CIDF so that event records, results and countermeasures could be shared (Tung 2000).

The Internet Engineering Task Force defined the Intrusion Alert Protocol (IAP) to allow the exchange of alert information between intrusion system elements. This was later improved in the Intrusion Detection Exchange Protocol (Buchheim, Erlinger et al. 2001) and published as a Request For Comments (RFC) (Debar, Curry et al. 2007).

NATO proposed a generic architecture for intrusion systems connecting trusted and untrusted network systems (North Atlantic Treaty Organization. Research and Technology Organization. 2002). This architecture included a similar number of elements as the CIDF, however they included visualisation and deception units.

For this thesis we have chosen to use an algorithmic framework. As such there are four generic models of intrusion systems, namely:

- Misuse Detection – where signatures are used to examine the available data. For host-based intrusion systems the signature can identify a specific virus or attack method, whilst for network-based systems they can detect the exploitation of application or operating system vulnerabilities;

- Anomaly Detection – in which resources are monitored to establish a norm for the systems under surveillance and the deviation from this norm is used to indicate an intrusion;
- Specification-based detection – where key processes or network protocols are monitored and specifications for correct behaviour of these processes are developed. For example in host-based intrusion systems, system call sequences could be monitored and any deviation from previously seen sequences is flagged as an intrusion; and
- User-based detection – In which normal user behaviour is determined and deviation from this behaviour is flagged as an intrusion. The behaviours that could be monitored include keystroke characteristics, application use and website access.

Specification-based and user-based detection can be considered as specific cases of anomaly detection. However they present different issues for algorithm designers and hence they have been separated here.

2.2. Data Processing

The focus of much academic research into improving intrusion system performance has been aimed at the selection of data processing algorithms. This section describes the most important data processing techniques that have been designed to improve discrimination between intrusion-like and non-intrusion-like activities.

2.2.1. Basic Techniques

A number of data processing techniques have been applied to the problem of detecting intrusions. They each attempt to address some of the fundamental issues, which include:

- Achieving high system confidence – to detect most (all) intrusions, with an acceptably low false alarm rate;
- Allowing performance optimisation – to select an arbitrarily high detection rate or an arbitrarily low false alarm rate, depending on the business impact of compromise of the information being protected or the ability to investigate false alarms;
- Adapting to changing system operating environments – to evolve in the presence of new, potentially uncharacterized intrusions, and in changing authorized user behaviour so as to maintain system assurance levels; and
- Surviving direct attack or evasion techniques – to maintain system assurance in the presence of attacks directed at the intrusion system or countermeasures aimed at evasion.

Intrusion systems operate as part of a set of system security tools whose aim is to achieve a defined level of assurance for the protection of the information assets accessed by the authorised system users. At one level, if each security element of a system were to operate correctly and effectively, then the need for high performance intrusion systems would be significantly reduced. For example, if network authentication systems were completely effective and not

subject to buffer overrun attacks, then the problem of intrusion detection would reduce to automated log analysis.

There are also network design approaches that can improve the ability of an intrusion system to operate effectively. For example in networks that incorporate darknet techniques (Qin, Dagon et al. 2004), in which IP addresses remain unassigned, it is trivial to detect intrusions that are attempting to access these addresses. Confident alarms can be generated with a detection probability of 1 for certain types of attack such as port or IP scanning. However, false negative performance can be poor if the intruder does not undertake reconnaissance or attacks on the darknet IP addresses. Intruders with detailed system knowledge, such as network support staff, could easily avoid these addresses. Also an intruder that has been passively monitoring network traffic using a network sniffer would not see activity on these addresses and therefore may not choose to attack using them.

From the preceding argument it is clear that intrusion system techniques should be able to alert to intrusion-like behaviours in the presence of normal behaviour, whilst making effective use of a priori network, user and threat information. It is in this context that the review of basic intrusion system techniques should be considered.

2.2.2. Support Vector Machines (SVM)

Support Vector Machines (SVM) have gained popularity as effective classifiers in a range of pattern recognition problems (Burges 1998; Muller, Mika et al. 2001). They are a generalised linear classifier, in which the input dataset is

partitioned into two classes using supervised training. They can be applied to N-class problems by combining SVMs in one of two ways. In one approach each SVM is trained to detect a particular class and reject all other classes. Such implementations are known as one-against-all SVMs. An alternative approach has SVMs trained to distinguish pairs of classes, where the total number of SVMs is the number of possible pairs of classes or N(N-1)/2. This is known as the one-versus-one approach and clearly the number of SVM classifiers can become very large for a large number N of classes.

SVMs have been applied to the problem of intrusion detection by a number of researchers (Burges 1998; Mukkamala, Janoski et al. 2002a; Quang, Zhang et al. 2002; Ambwani 2003; Fugate and Gattiker 2003; Hu and Heywood 2003; Hu, Liao et al. 2003; Ma and Perkins 2003; Mukkamala and Sung 2003d; Mukkamala and Sung 2003a; Mukkamala and Sung 2003c; Mukkamala and Sung 2003b; Quang, Zhang et al. 2003; Sung and Mukkamala 2003; Kim, Nguyen et al. 2005). Ambwani studied the use of one-versus-one method for multi-class SVMs applied to the KDD '99 dataset and to both problems of anomalous and misuse detection (Ambwani 2003). The detection performance was considered comparable to the winners of the KDD '99 competition.

Mukkamala (Mukkamala and Sung 2002; Mukkamala and Sung 2003d; Sung and Mukkamala 2003) systematically assessed the significance of the input features from the KDD '99 dataset, both for SVM and neural network based intrusion systems. They implemented a 5-class SVM using the one-against-all approach to classify the data into one of Normal; Denial of service; Remote

Access; Privilege Elevation; and Probing. High detection probabilities (>99.7%) were achieved for some classifications when the number of input features had been reduced to six from the original 41. Interestingly, Ambwani (Ambwani 2003) claims that the one-versus-one approach should provide better class discrimination that the one-against-all approach.

Chan studied the application of a hybrid SVM and rule-based approach to the detection of denial-of-service network attacks (Chan, Ng et al. 2004). An SVM was used to select important features within the data and to generate the rules. The rule-based sub-system was used to detect the denial-of-service attack. This work built on the work of Mukkamala (Mukkamala and Sung 2003c) and was able to demonstrate improved performance over systems where human experts had selected the ruleset, when applied to the KDD '99 dataset.

Nguyen used one-class SVMs as anomaly intrusion detectors (Nguyen 2002). One-class SVMs are unsupervised classifiers in which outliers (that is, anomalies) can be extracted. Connection-based features were extracted using TCPTRACE and applied to one-class SVMs trained to respond to each network service. High detection rate were achieved (100%) with false positive rates in the range 0.018-2.02% when trained and applied against the DARPA 1999 TCPDUMP dataset. Fugate (Fugate and Gattiker 2003) studied one-class SVMs as anomaly detectors, using the KDD '99 dataset.

Tran (Tran 2004) also studied one-class SVMs applied to the 1999 DARPA dataset. TCPSTAT was used to extract network statistics (features) over a defined time period, for input to the one-class SVM classifier. Only five of the

TCPSTAT parameters were used in this study. Despite this, detection rates of 71% were achieved with 20 false alarms per day.

Mill (Mill and Inoue 2004) extended the SVM architecture to include two new implementations. The training datasets were partitioned and used to train SVMs. In the TreeSVM implementation the results of the training on the first subset are used to train the next subset. A single set of SVMs result from this partitioning. In the ArraySVM implementation the SVMs that were trained on each partition were not combined, but instead placed in an array and applied to all the input data. The SVM which produced the greatest response to the input data-point was used to classify it.

Hu (Hu, Liao et al. 2003) applied SVM techniques to the detection of anomalies in host system calls and recognised that SVM performance is sensitive to the noise in the training data. Since it can be difficult to extract perfectly labelled data he proposed the use of Robust SVMs (RSVM) to overcome this limitation. Experimental results using the 1998 DARPA database demonstrated an improvement in detection performance compared with SVMs or K-nearest neighbour algorithms.

Kim (Kim and Cha 2004) applied SVMs to the difficult problem of masquerade detection in a host-based intrusion detection system. Sequences of user commands were used as the feature set and the results were compared with a naïve Bayes method. Detection rates of up to 87% were achieved with false alarm rates of 6.4%. Despite this poor performance the authors concluded that "*SVM is the most effective masquerade detection method available to date*".

Kim has applied this same approach to network masquerade detection in a web environment (Kim, Cho et al. 2004).

Kim (Kim, Nguyen et al. 2005) has used a genetic algorithm (GA) to improve further the performance of SVM intrusion classifiers. The GA subsystem was used to search for the optimal detection model, which was then evaluated by the SVM classifier. Results better than the KDD '99 competition winner were achieved.

SVM techniques remain a current topic of research due to their discrimination performance. Recent efforts have focussed on addressing implementation issues, such as the coarse-to-refined grid search techniques used during training SVMs by Lei (Lei and Zhou 2012).

2.2.3. Agents

The use of mobile agents to gather, process and take action using data distributed within a network has been extensively studied. Mobile agents (MA) would appear to offer many advantages over centralised models (Jansen, Mell et al. 1999) including:

- Reducing network load by processing data locally;
- Overcoming network latency by taking action at the infected host;
- Autonomous operation allows the intrusion system to continue operation as other parts of its implementation are attacked or destroyed;
- Platform independence, by allowing agents to interface to a variety of operating systems;

- Dynamic adaptation, using the mobility of agents to reconfigure an intrusion system in response to intruder activity;
- Static adaptation, by adding more agents as new threats or a priori data becomes available; and
- Scalability, as the computational load is spread rather than centralised.

Jansen (Jansen, Mell et al. 1999) also identified some limitations of mobile agents, including:

- Security – there are a large number of security concerns, including malicious MAs, attack by the MA host and eavesdropping attacks in transit;
- Performance – MA runtime environments are slow and can hinder the ability of intrusion systems to process events and detect attacks;
- Code size – The complexity of intrusion system tasks and the need to integrate with many different operating systems is likely to make the size of MAs large. This will increase the use of network resources;
- Lack of a priori knowledge – in large networks it is difficult to provide the MA with sufficient a priori knowledge of the network and its operating policies as to enable it to make accurate decisions;
- Limited exposure – MAs are not encountered frequently in operational networks and therefore there is a limited understanding of the issues posed; and
- Coding and deployment difficulties – there is a lack of design, development and management tools necessary to create and deploy

secure MAs.

Despite these limitations the mobile agent approach remains compelling as it has the ability to address the host and network intrusion domains simultaneously within a single paradigm.

A number of distributed intrusion system architectures have been studied, including GrIDS (Staniford-Chen, Cheung et al. 1996), NADIR (Hochberg, Jackson et al. 1993) and EMERALD (Neuman and Porras 1999). An early implementation of autonomous agents was studied by Balasubramaniyan (Balasubramaniyan, Garcia-Fernandez et al. 1998). Their approach, known as AAFID, uses agents as the lowest level of data collection and analysis and also includes a hierarchical structure to allow scalability.

Jansen studied the use of agents both for intrusion detection and response, recommending that further research is undertaken in three areas, namely:

- Intrusion system performance enhancement, through exploitation of the mobility of agents;
- New intrusion system design improvements, such as simultaneous detection; and
- Response improvements.

Mobile agents have been studied in mobile ad hoc networks (MANETS), in particular wireless networks (Hijazi and Nasser 2005; Xiao, Li et al. 2005; Sasikumar and Manjula 2012).

A recent survey by Kolias (Kolias, Kambourakis et al. 2011) on the application of swarm intelligence to intrusion detection evaluated 14 algorithms,

demonstrating detection performance comparable with the KDD '99 winner (see section 2.6.3.3).

2.2.4. Data Mining

Data mining has been defined as the process for the automatic extraction of models from large stores of data (Fayyad, Piatetsky-Shapiro et al. 1996). Lee (Lee and Stolfo 1998) studied its application to intrusion detection. SENDMAIL call sequences were mined to investigate host-based intrusion and network frames were mined for network intrusion application. An architecture for real-time model generation was developed demonstrating promising results. Manganaris (Manganaris, Christensen et al. 2000) investigated the mining of alarms generated from real-time intrusion detection sensors embedded in different networks. They were able to develop algorithms to associate false alarms and reject them depending on the underlying context of the alert characteristics of the network originating the alarm.

Julisch (Julisch and Dacier 2002) built on the mining of alarms, to develop insights into their root causes. In prior work he was able to show that alarm clustering was effective, demonstrating significant false alarm reduction when applied to alarms from real networks. Xiang (Xiang, Dong et al. 2005) have also developed algorithms to cluster alarms.

Data mining has also been used to augment the performance of discrimination systems. Lui (Lui, Fu et al. 2005) used three data mining techniques to provide features to an adaptive NIS consisting of five detection engines operating in an integrated manner. Jin (Jin, Sun et al. 2004) applied fuzzy data mining

techniques to network data, building on the work of Bridges (Bridges and Vaughn 2000).

In his recent review, Moorthy (Moorthy and Sathiyabama 2012) ascribes the principal application of data mining to anomaly rather than misuse detection. This was based on the inability of misuse detection to detect new, previously unseen attacks. Moorthy assessed the applicability of nine different classification techniques. Kaur has also surveyed data mining applied to intrusion detection (Kaur 2013) contrasting the relative merits of different techniques.

2.2.5. Artificial Neural Networks

The application of artificial neural networks (ANN) to intrusion detection is compelling and has been extensively studied. A recent review by Shah (Shah and Trivedi 2012) has compared five different network types applied to network anomaly detection. ANNs offer the ability to learn patterns in both a supervised and unsupervised manner, as well as to generalise from the exemplar patterns used during training.

Early research was based on their application within the IDEA intrusion system (Fox, Henning et al. 1990; Lunt 1990). Debar (Debar, Becker et al. 1992) used ANNs to model user behaviour from audit data. He linked a recurrent network with an expert system and was able to detect changes in user behaviour.

A number of ANN paradigms have been assessed in this application, including multi-layer perceptron (MLP) (Pan, Chen et al. 2003; Botha and Solms 2004; Cha, Vaidya et al. 2005) , radial basis functions (RBF) (Horeis 2003; Zhang and

Zhu 2004), self-organising maps (SOM) (Lei and Ghorbani 2004) and adaptive resonance theory (Di, Ji et al. 2005). Liu (Liu, Florez et al. 2002) studied the input representations for UNIX call sequence data, applied to back propagation, RBF and SOM networks.

Of particular interest is the work of Zhang (Zhang and Manikopoulos 2003) using the HIDE NIS (Zhang, Li et al. 2001). They have directly compared five ANN paradigms concluding that for denial of service attacks, the back propagation-hybrid and the back propagation networks outperform the others.

ANNs continue to be the subject of intense research for intrusion systems. Jahanbani has proposed the use of ANNs based on principal component analysis (PCA) as an anomaly intrusion system (Jahanbani and Karimi 2012). Gaikwad et al have combined fuzzy clustering with ANN using feed-forward networks (Gaikwad, Jagtap et al. 2012). Mamood et al have studied the application of ANN to cloud-based intrusion systems (Mahmood, Agrawal et al. 2012), using the back propagation algorithm and PCA pre-processing.

2.2.6. Fuzzy Systems

Intrusion systems are inherently quantitative, often relying on measurements of user, process, host or network activity. However the exact value of the measurements is often not important and this observation has stimulated the application of Zadeh's work (Zadeh 1988) on fuzzy systems, in this application.

The application of fuzzy techniques to data mining for intrusion detection has been the subject of considerable work with a number of researchers. Bridges (Bridges and Vaughn 2000) developed a prototype Intelligent Intrusion

Detection System (IIDS) to investigate the combination of fuzzy and genetic techniques. Fuzzy data mining was used to create membership functions for an anomaly detection system. Later Florez (Florez, Bridges et al. 2002), in collaboration with Bridges, extended this work to the creation of fuzzy association rules to compare recent audit data with previously mined "normal" behaviour. Tian (Tian, Fu et al. 2005) used fuzzy systems theory to combine decision trees applied to sub-sets of the complete mined database. They were able to show that this approach was superior to mining the complete database.

Dickerson (Dickerson and Dickerson 2000; Dickerson, Juslin et al. 2001) developed the Fuzzy Intrusion Recognition Engine (FIRE) to investigate the application of fuzzy systems to intrusion systems. Multiple agents were used independently to assess the situation and their outputs were combined via a fuzzy fusion algorithm. Each agent also applied fuzzy techniques to their input sources.

Gomez (Gomez and Dasgupta 2002) studied the use of fuzzy rule-sets to classify intrusions. The KDD '99 dataset was used to evaluate rules generated from a genetic algorithm. The results obtained were comparable to other techniques reported in the literature. Yao (Yao, Zhao et al. 2005) has investigated the use of fuzzy systems in the placing of decision boundaries, using an SVM as a classifier, also applied to the KDD '99 dataset.

Recent research has concentrated on the use of fuzzy methods in conjunction with other data processing techniques. Lei (Lei and Ke-nan 2011) investigated the application of fuzzy techniques in conjunction with SVMs and rough sets.

Experimental evaluation using a reduced set of parameters from the KDD '99 dataset indicated improved performance over the use of SVM alone. Ghadiri (Ghadiri and Ghadiri 2011) contrasted an improved version of fuzzy C-means with GK clustering as an input to radial basis function ANNs, again using the KDD '99 dataset. Ming-Yang (Ming-Yang, Chun-Yuen et al. 2011) studied the application of fuzzy association rules to anomaly-based network intrusion detection. Genetic optimisation of membership functions, yielded good detection performance for DoS attacks in synthetically generated network data.

2.2.7. Genetic Algorithms

The application of genetic algorithms to intrusion detection can be traced to Crosbie (Crosbie and Spafford 1995). They studied their use to create autonomous agents monitoring connections with a host. Me (Me 1998) extended this work applying genetic algorithms to misuse detection in host audit trails. Although good detection performance was achieved the technique was not able to locate the intrusion within the audit log. Gong (Gong, Zulkernine et al. 2005) studied the application of genetic algorithms using the 1998 DARPA dataset. He used only seven features of the network data to achieve effective discrimination.

Song (Song, Heywood et al. 2003) applied genetic algorithms to the KDD '99 dataset to create an anomaly based intrusion detector. He was able to show that the discriminator created after the evolution of the genetic algorithm was better than could be achieved when experts hand coded solutions. In later work (Song, Heywood et al. 2005) he showed that the approach was able to

create successful discrimination when the 41 parameters within the dataset were reduced to only eight.

Lu (Lu and Traore 2004) used genetic algorithms to adapt the discrimination rules for a network intrusion detection system. The DARPA dataset was used to evaluate the resulting performance. High detection probability and low false alarm rates were achieved even against attacks unseen in the training data.

Genetic algorithms continue to be an active research area for intrusion detection. Andhare has studied their use in denial of service detection (Andhare and Patil 2012). Boughaci uses genetic algorithms to update fuzzy "if-then" rules to enhance intrusion performance (Boughaci, Herkat et al. 2012).

2.2.8. Expert Systems and Probabilistic Reasoning

The application of expert systems to intrusion detection has a long history. The work of Denning (Denning 1987) created the Intrusion Detection Expert System (IDES) to process host audit files for anomalies. Later Anderson (Anderson, Frivold et al. 1995) extended this work to hybrid anomaly-misuse detection in the Next-generation Intrusion Detection Expert System (NIDES). EMERALD also used a hybrid detector, this time to process host and network data (Neuman and Porras 1999). Lindqvist (Lindqvist and Porras 1999) developed the expert system engine known as P-BEST, a key component of the EMERALD system.

Many researchers have taken a probabilistic approach to intrusion detection. Seleznyov (Seleznyov, Terziyan et al. 2000) used probabilistic trees to encode the temporal behaviour of users and then detect anomalies. Ye (Ye, Li et al. 2001) studied the probabilistic properties of audit host data, concluding that

multiple events were necessary before intrusion could be declared with confidence. Leckie (Leckie and Kotagiri 2002) studied the application of probabilistic techniques to the processing of network data to detect port scans. Gowandia (Gowadia, Farkas et al. 2005) has studied the application of probability theory to agents, developing a means for them to share their beliefs.

The majority of workers have taken a Bayesian approach to the application of probability theory. Alternatives based on the theory of evidence have been applied by Chen (Chen and Venkataramanan 2005) who studied the application of Dempster-Schafer (D-S) theory to intrusion detection. D-S theory appears to offer the advantage of combining data from sources with differing levels of trust. This is particularly important when the sensors providing data to the intrusion system are distributed within the network and may have been compromised by an intruder. Interestingly the application of possibility theory (see (Borotschnig, Paletta et al. 1999), for example) does not appear to have been applied to problems in intrusion detection, except within the context of fuzzy systems.

2.2.9. Digital Signal Processing (DSP)

The application of DSP techniques to network intrusion systems has been extensively studied. Whilst the frame data on networks is asynchronous, it is easy to generate time-series data by, for example, deriving statistics over fixed time periods. Thottan (Thottan and Ji 2003) used MIB data accessed via SNMP, and applied time-series processing. They were able to predict network equipment failure before complete failure had occurred. The use of non-uniform

sampling algorithms from DSP does not seem to have been applied to intrusion systems.

Axelsson (Axelsson 2000b) attempted to apply classical detection theory to intrusions. He was able to assert that different attack methods mapped on to different classes of problems from classical detection theory. For example, masquerading was thought to be equivalent to "detection of random signals in random noise". Unfortunately, he was unable to apply this observation to aid the detection of intruders.

Barford (Barford, Kline et al. 2002a) generated time-series data from SNMP and from IP flow data. He applied a number of time-frequency analysis techniques and found that wavelets were able to isolate both short and long duration traffic anomalies. Zhou (Zhou and Lang 2003) created time-series data from the number of frames arriving in unit time and then studied the use of frequency based techniques using the discrete fourier transform.

AsSadhan created time series data by aggregating data over selected time periods (AsSadhan 2009). He aggregated frames, bytes, distinct addresses and distinct ports and was able to show that by analysing the control and data planes for TCP-IP connections anomalous behaviour could be detected. The cross-correlation function was used to measure the similarity between these planes and low similarity was used to indicate malicious behaviour. In addition he was able to detect period characteristics using frequency domain techniques, which could be related to botnet command and control communications.

Digital signal processing techniques remain an open area for research into

intrusion detection.

2.2.10. Miscellaneous Techniques

Game theory has been applied to the problem of detecting intrusions by a number or workers. Alpcan (Alpcan and Basar 2003) assessed the activities of intruders and intrusion system control strategies as a two-person non-zero sum, non-cooperative game. Kodialam (Kodialam and Lakshman 2003) studied the use of game theory to detect intruders in sampled network data. Agah (Agah, Das et al. 2004) also used two-person, nonzero-sum, non-cooperative game between the intrusion system network and the attacker. They showed this approach significantly improved the chances of intrusion detection. Patcha (Patcha and Park 2004) extended Alpcan's work to MANETS defended by host intrusion systems. Rafsanjani has studied the application of game theory to intrusion systems for MANETS (Rafsanjani, Aliahmadipour et al. 2012).

Aickelin (Aickelin, Bentley et al. 2003) studied the application of danger theory to immunological intrusion system. Danger theory is emerging as an alternative to self– non-self determination as a model of the human immune system. Aickelin proposed the use of danger theory to associate low-level detection as an alternative to probabilistic or expert system correlation. Lu et al studied the application of danger theory to mobile virus detection (Lu, Zheng et al. 2012).

He (He and Leung 2004) studied the application of chaotic stochastic resonance to intrusion detection. A simplistic approach to intrusion was taken in which an anomaly was declared when the difference between the predicted and the actual frame size exceeded a threshold. Of particular interest was the setting of

the threshold, in which a constant false alarm (CFAR) method was used. This was the first application of CFAR techniques to intrusion detection. However, the assumption of stationary noise statistics resulted in a fixed threshold, limiting the usefulness of this research.

Hidden Markov Models (HMM) have been successfully applied to many pattern recognition problems. Gao (Gao, Ma et al. 2002; Gao, Sun et al. 2003) studied the application of HMMs to anomaly detection in UNIX process call sequences, with good results. Zhang (Zhang and Zhu 2004) combined HMMs with ANNs claiming improvements over HMM techniques used alone. These improvements were mainly in terms of storage and processing requirements. The effect on detection performance of this combination was unclear. More recently HMMs have been used to extract the interaction between intruders and network devices, to predict multi-stage attacks and prevent further damage (Shameli Sendi, Dagenais et al. 2012)

Petri nets have also been applied to the problem of intrusion detection. Ali (Ali 2001) modelled the monitoring function of the CIDF using Petri nets. Helmer (Helmer, Wong et al. 2001) continued the theme and used Petri nets to model the specification for an agent-based intrusion systems. Gao (Gao and Zhou 2003) used Petri nets as part of the discrimination function, by using them to encode fuzzy rules and to make the intrusion/non-intrusion decision.

Intrusion systems based on user behaviour has been extensively investigated (Kakuru 2011; Razo-Zapata, Mex-Perera et al. 2012). Users can be authenticated based on the frequency with which they use console commands

or applications, in host-based intrusion systems, and the websites accessed in both a host and network-based intrusion system. Feher (Feher, Elovici et al. 2012) investigated mouse movements to authenticate users. Despite issues with user characteristics depending on the location of the user and specifically the input devices available, excellent results were obtained.

2.3. Post Intrusion Processing

An intrusion system generates alerts information that can be further processing in a number of ways:

- Local logs, in which the alerts are stored within the intrusion system and are not integrated with other log sources within the enterprise. For distributed approaches, such as NIS, the alerts can be integrated from several intrusion sensors centralised into a single store; and
- Enterprise logs, in which the alerts are combined with logs from other, heterogeneous devices to provide a centralised view of the security of the complete infrastructure. Such approaches are known as Security Information and Event Management (SIEM) and can provide a further level of intrusion detection based on the events seen at other non-intrusion devices in the enterprise.

The simplest form of local logs occurs when a single intrusion sensor is monitoring a resource, such as a network intrusion system monitoring the Internet gateway, or anti-virus software monitoring a single operating system. Analysis of the logs is usually undertaken by system administration staff accessing the log files directly, or undertaking an automated analysis using bespoke scripts designed to assess specific concerns. Tools such as SGUIL are

available to assist in these tasks (Visscher 2007) for network intrusion systems.

Enterprise log handling presents different issues from local logs, as summarised in the guide to computer security log management (kent and souppaya 2006) produced by the National Institute of Standards and Technology (NIST). They identified the fundamental problem of log handling to be the matching of the limited log management resources with a continuous supply of log data. They identified additional challenges including:

a) Large number of log sources within an enterprise;
b) Inconsistent log content, formats and timestamps; and
c) Protecting large quantities of log data, whilst allowing access to system administrators.

In an attempt to standardise on the communications between intrusion systems the Internet Engineering Task Force (IETF) proposed the Intrusion Detection Message Exchange Format (IDMEF) in RFC4765 (Debar, Curry et al. 2007). Two approaches to the remaining issues have been extensively studied, namely methods for correlating logs and visualisation techniques.

2.3.1. Correlating Logs

Abad et al undertook early work assessing the ability to correlate intrusion logs (Abad, Taylor et al. 2003). There were able to demonstrate improved accuracy of intrusion detection by correlating system calls and network logs using a sliding window. Valeur defined a comprehensive approach to alert correlation (Valeur, Vigna et al. 2004) proposing a nine stage process rather than limiting it to just a few stages. Although depicted as a serial process with each stage

applied sequentially, Valeur allowed some of the stages to occur in parallel, also including feedback between stages.

Elshoush has surveyed alert correlation applied to an intrusion system consisting of cooperative misuse and anomaly approaches (Elshoush and Osman 2011). She used the five correlation techniques assessed by Xu (Xu 2006), to process the output from discrete intrusion systems communicating via IDMEF data format, specifically:

- Similarity between alerts;
- Pre-defined attack scenarios;
- Prerequisites and consequences;
- Multiple sources; and
- Filtering.

Feng et al used state machines to correlate security events (Feng, Wang et al. 2010). Attack scenarios were reconstructed using state machines combining clustering and causal analysis to output to a comprehensive description of the attack.

Jing has evaluated the use of rough sets as a data reduction technique for event correlation (Jing, Lize et al. 2012). A pattern mining algorithm was then applied to generate the correlation rule without using prior knowledge. Although there is a reduction in data rate using rough sets the false alarm rate increased.

More recently Salah et al have undertaken a survey of alert correlation techniques (Salah, Macia-Fernandez et al. 2013). This approach described the

state-of-the-art in alert correlation, not limited to computer systems but also industrial control. A taxonomy for correlation was proposed, based on number of data sources, type of application, correlation method and type of architecture. A review of commercial solutions was also given where they identify the problem of no clear agreement between researchers and vendors about the performance metrics. They note that there are no standard benchmarks for evaluating and comparing such systems.

2.3.2. Visualisation

One of the problems facing network security staff is the quantity of information contained in network scans and logs. For example, a single 100Mb link may well transmit 10^{10} bits during a single day. Some researchers have proposed the use of visualisation techniques to aid intrusion detection by network staff. Teoh (Teoh, Ma et al. 2002; Teoh, Ma et al. 2003; Teoh, Ma et al. 2004) investigated the use of visualisation for Border Gate Protocol (BGP) routing anomaly detection. They have also applied their techniques to intrusion systems (Teoh, Ma et al. 2004) with the result that have exceeded the performance achieved by the winners of the KDD '99 competition.

Conti (Conti and Abdullah 2004) investigated the visual signatures produced by network attack tools and was able to demonstrate that a number of visualisation techniques could be effective. The port-to-port plots were particularly good at fingerprinting tools such as NMAP and NIKTO. Conti recognised that the availability of source code for many of these tools would allow an attacker to modify the signature and potentially evade detection using visualisation techniques.

Koike visualised the output from SNORT using the SnortView tool (Koike and Ohno 2004). Visualisation was proposed as an alternative to optimising the signature database, as SnortView was able to highlight both true and false alarms, based on a time series view of the intrusions.

Axelsson (Axelsson 2005) analysed four different visualisation techniques applied to web server logs. He investigated both the visualisation of network data and the visualisation of the internal status of intrusion detection systems. He concluded that the topic was immature and lacking effective user studies.

Livnat (Livnat, Agutter et al. 2005) investigated visualisation for associating data from disparate system logs, defining an alert in terms of what he called the W3 paradigm, meaning What, Where and When. Efficient ways of presenting data in terms of these parameters were developed, which are scalable to large networks. Interestingly, they did not investigate the pre-processing of the data into intrusion system relevant parameters, but chose to present the basic log data.

Yang developed a visualisation approach for network alerts (Li, Gasior et al. 2010). Visualisation of the network topology was combined with visual clustering of alerts to alert users. A third stage was included to view multistage attacks by temporally profiling user and attacker behaviour.

More recent research into visualisation has concentrated on SIEM, which integrates additional information from other network devices and servers. Novikova has proposed a visualisation framework based on a service-orientated approach (Novikova and Kotenko 2013) demonstrating the concept applied to

the attack modelling and security evaluation component of an SIEM (Kotenko and Chechulin 2012). Xiaojin has developed a tool known as VisSRA to visualise both the rules and the alerts from SNORT using treemaps (Xiaojin, Changzhen et al. 2012).

2.3.3. SIEM

Within an enterprise network it is common for the output of intrusion systems to be sent to an SIEM application, for integration with logs from other systems, such as firewall, routers and security devices. At their simplest level SIEMs collate enterprise wide logs, providing support tools to aid the interpretation and management of events. Many of these tools use visualisation or log correlation techniques to assist users in understanding the status of their networks. Gartner has produced a "magic quadrant" analysis (Nicolette and kavanagh 2011). This analysis considers 25 commercial systems and describes the market as "mature and competitive". This is in contrast to previous studies, such as Shipley, (Shipley 2008) which identified short-comings in the usability, reporting and event correlation aspects. Interestingly, Shipley reported that over 6,000 SNORT alerts were rejected by the Q1 Labs QRadar SIEM, which chose only to display one alert as valid. The same SIEM reported a 500,000:1 data reduction for logs as a result of the analysis and correlation of events.

SIEM systems are becoming of interest to academic research. Gabriel used data mining to detect hidden patterns in malware data within an SIEM (Gabriel, Hoppe et al. 2009). Kotenko has proposed a common framework for the Attack Modelling and Security Evaluation Component (AMSEC) of a SIEM (Kotenko and

Chechulin 2012). Their approach integrates open source vulnerability databases, such as CVE, with near real-time attack modelling to predict the future actions of an attacker. Kotenko has also studied the issues surrounding the data repository of an SIEM, proposing an ontological approach (Kotenko, Polubelova et al. 2012). An ontological data model of vulnerabilities was discussed for an AMSEC. Granadillo et al have also used an ontological approach to SIEM modelling, applied to botnets (Granadillo, Mustapha et al. 2012).

Afzaal has assessed some of the systems aspects of the use of SIEM, proposing a resilient architecture for forensic storage of data (Afzaal, Di Sarno et al. 2012).

2.4. Intrusion Systems

The techniques discussed in section 2.2 form the discrimination or decision function within an intrusion system. Other functions necessary to form a complete system can include:

- Data pre-processing, in which the raw information is transformed such that the discrimination function sees a consistent dataset. De-fragmenting frames is an example of pre-processing, in which a number of frames are combined to create (usually) larger frames;
- Alert logging, which enables the details of an alert to be stored and retrieved in meaningful ways. A simplistic approach to logging can be taken, in which the raw data is stored along with some information from the discriminator, in a simple file structure. An alternative, more complex

approach can be taken in which the alerts are inserted into a database for retrieval and analysis offline, by other tools;

- Configuration Management – An intrusion system can be tuned to respond to different threats, via the selection of discrimination techniques, decision thresholds or rule-sets. It is necessary to provide tools to manage the editing, display and logging of the configuration of the intrusion system;
- Threat Analysis – A single alert from an intrusion system can sometimes be definitive on the presence and nature of an intrusion. However, usually a single alert cannot provide the confidence to declare an intrusion without the presence (or absence) of other alerts or information. To this end, intrusion systems need to provide tools which allow the context of an alert to be assessed. Frequently, these analysis systems involve data visualisation techniques, as described in the previous section;
- Response Systems, in which the presence of an alert triggers an action from the intrusion system. Actions can include dropping frames, updating firewall rule-sets or blocking specific IP addresses; and
- Protection tools – The extent of the security functionality that can be provided by intrusion system is such that they can become a prime target for intruders. It is therefore necessary to provide a suite of protection tools to ensure that the intrusion system cannot be compromised. Such tools can include encryption mechanisms to hide the details of the communications with a centralised controller, data rate

throttling mechanisms to deal with denial of service attacks or interface modification to prevent the transmission of data from the intrusion system interface.

Although the functionality listed above is necessary to create a complete intrusion system there appears to be a lack of systematic study of the interaction of this functionality with the discrimination function. There are instances where the discrimination function is highly dependent on the additional functionality. For example the use of visualisation techniques applied to the raw network or host data is closely related to the threat analysis functionality.

Despite this lack of systematic study a number of research organisations have constructed complete systems and evaluated their performance. There have been extensive published reviews of research on such intrusion systems (Allen, Christie et al. 1999; Axelsson 1999b; North Atlantic Treaty Organization. Research and Technology Organization. 2002) and therefore this section will only indicate some of the key issues.

2.4.1. Research Systems

Axelsson (Axelsson 1999b) reviewed twenty research systems in terms of key parameters such as:

- Detection principle;
- Real-time or non-real-time operation;
- Continuous or batch operation;
- Network or host data sources;

- Passive or active response;
- Centralised or distributed processing;
- Centralised or distributed data collection;
- Security; and
- Interoperability.

This survey provides significant insight into the modes of operation of complete systems, although it is now becoming dated with respect to the systems assessed.

One of the most popular signature-based NIS is SNORT (Roesch 1999), developed as an Open Source project. SNORT is able to detect misuse through the application of user-defined rules. Pre-processors are included to normalise the captured frames and to extend the intrusion criteria over more than one frame (Caswell, Beale et al. 2003). Although originally thought to be a lightweight intrusion detection system, it is part of commercial products and has been a core component of many research systems.

2.4.2. Commercial Systems

There are a number of commercial network intrusion systems. Several are based on the SNORT detection engine augmented with network characterisation techniques to reduce initial deployment false alarm issues or additional tools to improve operator productivity. It is difficult to review these systems due to the commercial sensitivities, however some researchers have attempted this problem (Allen, Christie et al. 1999; Kvarnstrom 1999). Debar (Debar and Morin 2002) evaluated some commercial systems without

identifying their type. Evaluations of more modern commercial NIS appear to be lacking.

2.4.3. Systems Approaches to False Alarm Control

Although the research emphasis for reducing false alarms has been on data processing techniques, a number of systems approaches have also been proposed. Shimamura (Shimamura and Kono 2006) proposed that false alarms can be reduced from NIS if alerts that have no effect on the system are ignored. For example if a particular attack is launched against a specific OS and that OS is not present in the system then the attack cannot damage the system and the alert can be suppressed. During an evaluation of 15 days of real network traffic the system, known as TrueAlarm, reduced the number of false alarms from 125 for a conventional NIS to zero. Whilst this result appears to be good it does suffer from a number of issues:

- The assertion that an attack that cannot damage a system is of no interest to administrators is flawed. Such attacks can occur, for example, during the reconnaissance phase when an attacker is learning about the services provided by a network. Although some reconnaissance probes may not affect the system the presence of persistent attempts to enumerate, penetrate or damage a system is an important indicator to administrators and may allow measures such as IP blocking to be deployed before the damaging attack occurs;
- The analysis does not address the impact on detection probability. For TrueAlarm this is problematic as non-damaging attacks should be

prohibited by the network security policy and therefore be flagged as network intrusions. Whilst it is conceivable that the network security policy could be relaxed to maintain the detection probability, the principle of relaxations in security to overcome system performance issues is flawed; and

- Since the alert occurs after the attack frame has delivered its payload to the server, this approach cannot be used in-line or in intrusion prevention systems

Bolzoni (Bolzoni and Etalle 2006) proposed a novel approach to reducing false positives in a system known as Aphrodite. His approach was to correlate the output of a NIS placed on the incoming network stream with anomalous responses from the attacked system. A separate anomaly detection system was placed on the output from the attacked system and an alert is declared when both the NIS and anomaly detection alert. False positive reductions of between 50-100% without affecting detection rate were reported. Although this performance improvement is significant the attack must have to be successful to be detected, that is the system must be adversely affected and an anomalous output produced. This is unsatisfactory and limits the applicability of Aphrodite in IPS scenarios.

In host-based intrusion a systems approach has been particularly successful. Kim (Kim and Spafford 1994) described a file integrity based approach known as Tripwire. An alert is issued when any of the key system files are changed on a host. This approach is highly successfully and is included in a number of commercial and Open Source systems.

Hofmeyr (Hofmeyr, Forrest et al. 1998) investigated the use of short sequences of system calls to discriminate between normal and abnormal conditions for common UNIX programs. This idea was successfully extended to Microsoft Windows environments by looking for unusual registry accesses (Apap, Honig et al. 2002; Topallar, Depren et al. 2004). Host intrusion detection for mobile devices has also examined the feasibility of correlating abnormal battery behaviour with intrusion events (Buennemeyer, Munshi et al. 2007).

More recently a systematic attempt to quantify false alarms has been published (Cheng-Yuan, Ying-Dar et al. 2012; Cheng-Yuan, Yuan-Cheng et al. 2012). In this study over 2,000 instances of false positive and negatives were identified in real network data taken over a 16 month period. Given that the volume of network traffic is quoted as 100GB/hour this rate of false alarms seems remarkably low. Nearly 93% of the false alarms were classified as false positives, with the majority of those, in their view, not being the result of security issues but the result of security policy violations. From their published methodology it is difficult to determine that all the false alarms have been counted, in particular the false negatives. As the quantity of network data is so large manual confirmation of missed detections by inspection of individual frames is not possible. However they do not seem to have chosen to inject known intrusions to confirm their measurements.

2.4.4. Intrusion System Limitations

Axelsson (Axelsson 1999a) highlighted an important limitation of intrusion systems. He correctly recognised the problem posed by the very small ($\sim 10^{-6}$) a

priori probability of an intrusion event in large datasets. Under such conditions false positives can be eliminated by labelling every frame as not resulting from an intrusion. Clearly this is unhelpful in detecting intrusion events, but the rate of false negatives is small due to the small a priori probability.

2.5. Intrusion Evasion

As the deployment of intrusion systems has increased so has the interest in evading them. Both the academic and the intruder communities have developed tools and techniques to evade detection by intrusion systems. The principal academic work was undertaken by Ptacek (Ptacek and Newsham 1998). As well as describing denial-of-service attacks on intrusion systems, this widely cited paper described two techniques to avoid detection through confusion of the sensors, namely:

- Insertion – In which frames destined for the target host are seen by the intrusion system but not seen by the target host (for example they could be rejected by the host due to checksum errors); and
- Evasion – In which the intrusion system is fooled into rejecting some of the frames that the target host correctly processes.

Both these techniques rely on the intrusion system processing the frames in a different way to the target host. One way to overcome this problem is to include a traffic normaliser within the network, as proposed by Handley (Handley, Paxson et al. 2001) and used in SNORT (Caswell, Beale et al. 2003).

A number of Open Source tools have been created using these techniques. NESSUS, the well-known vulnerability scanner, has intrusion system evasion

techniques included within its architecture. SNOT is an arbitrary frame generator that uses the SNORT rules files as a source of frame information. It has been used in intrusion system testing, as an evasion tool and to overload IT security staff with false alerts. STICK is a NIS stress tool designed to generate a large number of alerts and cause the intrusion system to crash. WHISKER is also aware of evasion techniques.

Other techniques to avoid intrusion systems include:

- Encrypting the data, which can hide signature information from misuse detection systems;
- Slow probing, in which the network probes are send with a large delay between them. The aim is to get the intrusion system to timeout;
- Fragmentation, to hide the signature of the attacking frames; and
- False-alarm attack, in which the intrusion system is tricked into generating a large number of alarms, so as to hide the real intrusion alarms (Patton, Yurcik et al. 2001).

Fogla has studied the evasion of anomaly network intrusion systems (Fogla and Lee 2006). He proposed that such systems could be evaded by a Polymorphic Blending Attack (PBA), in which the statistics of an attack match those of normal network traffic. He demonstrated that PBA attacks could be generated automatically for arbitrary code execution and showed that anomaly intrusion systems could be evaded.

Pastrana et al have proposed a functional framework for evading network intrusion systems (Pastrana, Orfila et al. 2011). They used genetic

programming techniques to derive a model of the intrusion system, from which evasion strategies were derived. Evasion of SYN flooding and port scanning attacks was demonstrated.

Tran et al have studied the evasion of SNORT by exploiting flowbits (Tran, Aib et al. 2012). They parse the active signature set running on SNORT to generate all possible frame sequences that can evade it. They also show that additional signatures can be added to overcome the evasion.

More recently Cheng et al have evaluated evasion techniques against intrusion systems (Cheng, Lin et al. 2012). They compared five common evasion techniques against three different signature based network intrusion systems, including SNORT. All three systems could be evaded using IP fragmentation and TCP segmentation.

2.6. Evaluation of Intrusion Detection Systems

There has been considerable research into the experimental evaluation of intrusion systems. A number of approaches have been used, including:

- Evaluation on live networks or hosts;
- Evaluation against standard databases of network, host and intruder activity;
- Generation of synthetic network or host data; and
- Development of performance metrics.

The aim of any evaluation must be to allow accurate comparison of systems or intrusion classification techniques in such a way that further research can be focused. Live networks can generate qualitative data but it is difficult to

compare systems directly without their operation in parallel. Performance metrics offer a quantitative approach and can be particularly effective at comparing systems. It is important however that the basis for calculation is common and that the metrics are calculated in a way that allows comparison. This usually requires the use of a standard labelled database of normal and intrusion behaviour.

It is possible to consider the collection of network frames or host data from real networks and disseminate the resulting database to researchers for evaluation using common data. This approach has two problems:

a) It would be difficult to label the intrusions in such a way that learning algorithms could be trained using noise free data. It would be possible to launch attacks on the network elements and generate correctly labelled attack data. However, at the times in which deliberate attacks are not being undertaken it would be difficult to guarantee that malicious activity was not being undertaken by system users or by real intruders that had penetrated the system; and

b) Data from a real network will have sensitive information contained within its frames. The presence of personal information may restrict the data that could be released and the detailed network information, such as server addresses, may assist a future intruder to penetrate the network. Whilst there are techniques for changing the data such that these issues could be overcome, these changes may adversely affect the training or evaluation of intrusion systems.

As a result, the assessment of intrusion systems is dominated by the use of data generated from synthetic networks (the DARPA datasets), despite its known limitations.

2.6.1. Security Metrics

The application of security metrics to quantify the state of a computer system has been extensively studied. Almasizadeh assessed a state-based stochastic model of the attack process to represent the security of a system, from which quantitative metrics could be derived (Almasizadeh and Azgomi 2013). Two specific metrics were assessed namely the mean time to a security failure and the steady state solution describing the long-term state of the system, that is, the likelihood that the system is in each state.

Ouedraogo et al have studied the application of security assurance metrics to operational systems, rather than the development process as in Common Criteria (Ouedraogo, Khadraoui et al. 2012). The security of a system is assessed into one of five levels using probes, based on the Systems Security Engineering Capability Maturity Model (SSE-CMM).

Bayuk has reviewed the evolution of security metrics from the technical and historical perspective (Bayuk and Mostashari 2013). She concludes that metrics are sufficient for verification of a system, that is, it is built and operated to a specified level, but that validation metrics are less mature. Bayuk used cloud security as an example of the application of her metrics (Bayuk 2011).

Chrun studied the use of an intrusion prevention system to derive security metrics (Chrun, Cukier et al. 2008). He proposed and measured ten metrics

equally split between outside and inside threats, using live data from a system with 40,000 users. The metrics were produced as time-series data, using a sliding window for their measurement. Despite the known limitations of signature-based intrusion system a major security incident was detected using these metrics, which had failed to be detected via the normal operational security techniques in place within the organisation.

Lippmann et al have recently proposed four metrics for network security threats (Lippmann, Riordan et al. 2012). Using the twenty critical controls for effective cyber defence defined by SANS (SANS 2013), fifteen are able to be continuously measured.

2.6.2. Performance Metrics

There are many parameters of intrusion systems that could be measured so as to characterise performance. These include:

- CPU load;
- Memory requirements;
- Latency of detection declaration;
- Maximum network data rate; and
- Protocols that can be assessed.

Whilst such metrics can be useful it is the ability of an intrusion system to detect intrusions whilst giving an acceptably low false alarm rate that is the most important, and the most challenging issue.

A review of some of the detection metrics is provided by Abouzakhar (Abouzakhar and Manson 2004). The ability of an intrusion system to detect

intrusions is often characterised by its detection probability, P_d defined as the ratio of the detected intrusion events divided by the total number of intrusion events. Sometimes detection rate rather than P_d is used, in which the ratio is expressed as a percentage.

The false alarm performance is often characterised by the probability of false alarm, P_{fa}, which is the ratio of the number of false alarms declared divided by the total number of possible false alarm events that could be declared. Again false alarm rate is often used, which is P_{fa} expressed as a percentage.

It should be noted that P_{fa} measures the false positive instances and therefore does not represent the full characteristics of false alarms. It does not include false negatives, in which a real intrusion event is missed. Therefore some researchers use confusion matrices to fully record the true positive (TP), true negative (TN), false positive (FP) and false negative (FN) statistics of an intrusion system.

A frequently used assessment method is to plot P_d versus P_{fa} (or detection rate against false alarm rate). The resulting graph is known as a Receiver Operating Characteristic (ROC) curve. ROC curves allow an intrusion to be assessed at different sensitivity settings (Fawcett 2003). Some researchers use a modified ROC curve, known as a lift curve, which has similar properties to a ROC curve (Abouzakhar and Manson 2004).

Another measure frequently used is precision (Pr), which is defined as

$$\Pr = TP/(TP + FP) \times 100\%$$

Equation 2-1 Definition of Precision

Precision is often calculated with another parameter known as accuracy (Ac) which is defined as:

$$Ac = (TP + TN)/(TP + TN + FP + FN) \times 100\%$$

Equation 2-2 Definition of Accuracy

The aim is to get both precision and accuracy to approach 100%, which can be achieved by zero false positives and zero false negatives.

It is important to undertake the comparison of intrusion systems using the same basic assumptions. For example, whilst ROC curves for different systems can be compared directly this should not be done without detailed consideration of their methods of calculation. Consider the following situation. An intruder has found a way into a network that is monitored by a NIS. The intruder sends ten frames to another host in order to penetrate further into the network. The NIS detects one of the frames and initiates an alert. What is the P_d for the NIS in this situation? Since only one frame in ten was detected the P_d could be characterised as 0.1 (the per-frame view). Alternatively, since the activities of the intruder in attempting to penetrate the next host were detected the P_d could be considered to be 1 (the per-connection view). If some of the frames used by the intruder were legitimate, for example a ping, then they could not be expected to be detected and the resulting P_d could therefore be somewhere between 0.1 and 1.0 (the per-malicious frame view). Thus, depending on the underlying assumptions, significantly different ROC curves can be created under

identical intrusion conditions. This is discussed again in section 4.6.

The problem of comparison of different systems can be seen by examining Figures 2-1 and 2-2. Figure 2-1 was taken from Lippmann (Lippmann and Cunningham 2000) and shows the comparison of three different intrusion systems. It is easy to see that the "Neural Net" approach is better than either the new or old keyword count methods as at any given false alarm per day the attacks detected are greater.

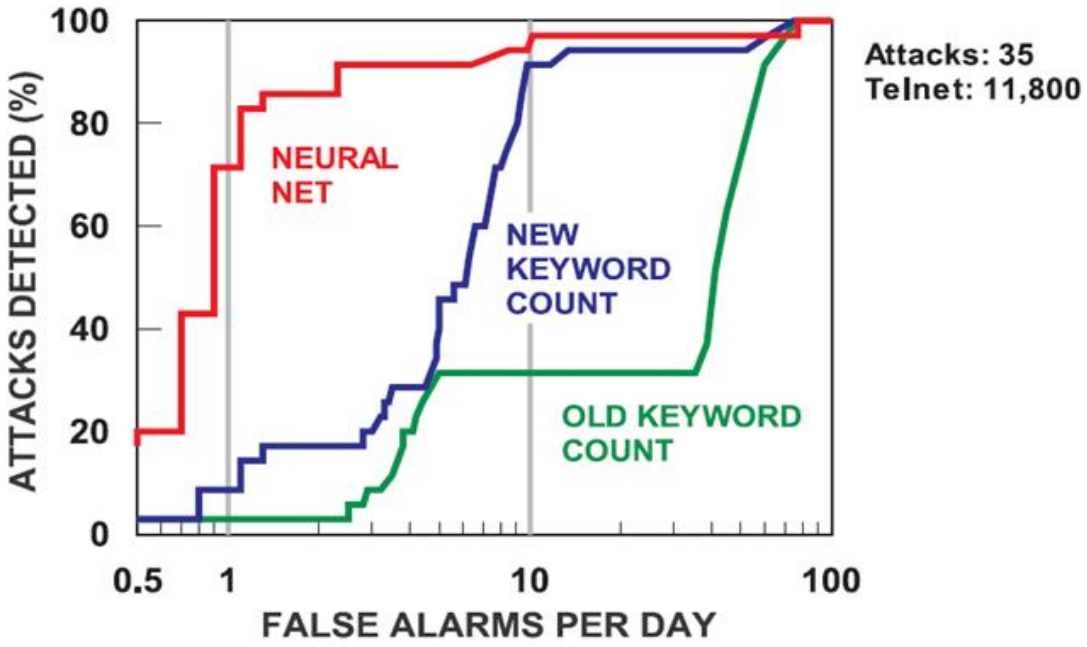


Figure 2-1 Intrusion System Performance (Lippmann and Cunningham 2000)

Consider now figure 2-2 taken from Estevez-Tapiador (Estevez-Tapiador, Garcia-Teodoro et al. 2004). Is the intrusion system produced by Estevez-Tiador better than any of the systems investigated by Lippmann? Both research teams label these diagrams as ROC curves but direct comparison is extremely difficult.

During the literature research phase of this thesis 62 ROC curves were collected from the published research of 22 teams. From this it was clear that direct comparison between research groups was extremely difficult, but within groups it was frequently straightforward. The lack of an agreed and uniform way of publishing quantitative performance measurements does not help identify research opportunities.

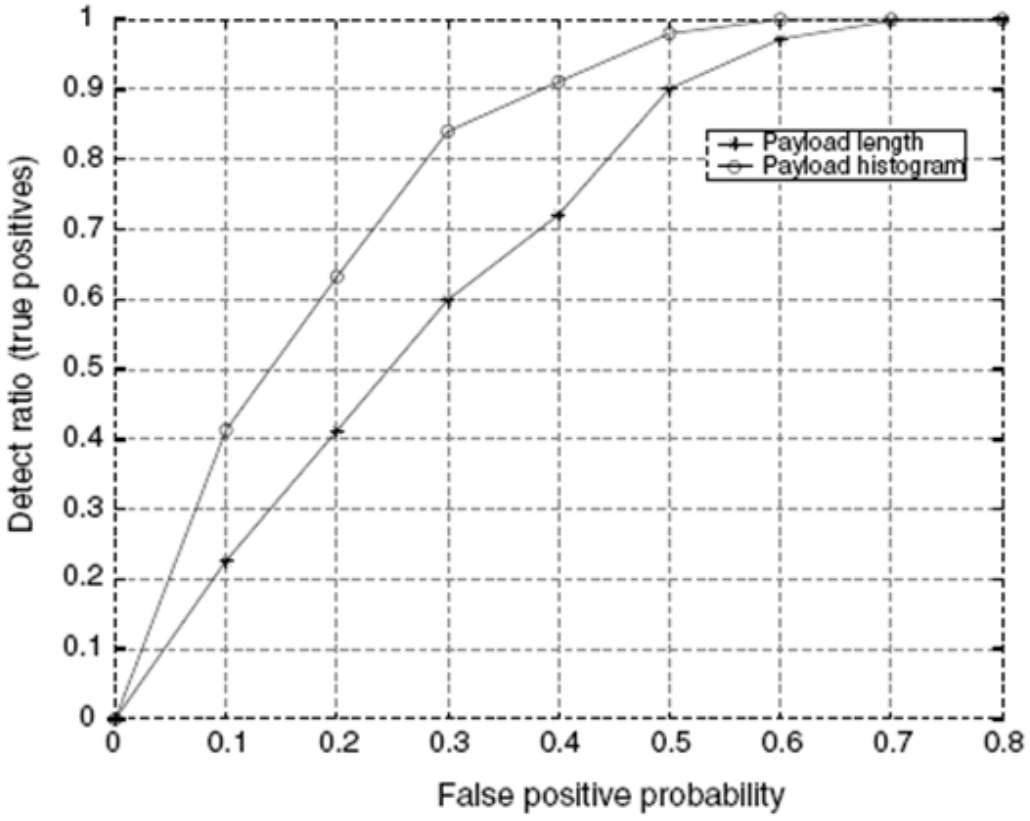


Figure 2-2 ROC Curves (Estevez-Tapiador, Garcia-Teodoro et al. 2004)

Alternative ways of defining and measuring performance are discussed further in Chapters 3 and 4 of this thesis.

2.6.3. Activity Databases

A number of databases have been produced to assist in the evaluation and comparison of intrusion systems. These databases are important as they

potentially allow direct comparison of the techniques used by different researchers. They are in widespread use.

2.6.3.1. The 1998 DARPA Dataset

In 1998 DARPA, in conjunction with MIT Lincoln Laboratory conducted an evaluation of intrusion systems. An evaluation test bed was developed which simulated a government network of 100's of users on 1000's of hosts. More than 300 attacks of 38 different types were launched against UNIX hosts during seven weeks of training and two weeks of testing (see Table 2-1 adapted from (Lippmann, Fried et al. 2000)).

	Solaris	SunOS	Linux	Cisco Router
Denial of Service (DoS)	apache2 back mailbomb neptune process table ping of death smurf syslogd udp-storm	apache2 back land mailbomb neptune ping of death process table smurf udp-storm	apache2 back mailbomb neptune process table ping of death smurf teardrop udp-storm	
Remote to Local (R2L)	dictionary ftp-write guest http-tunnel phf xlock xsnoop	dictionary ftp-write guest phf xlock xsnoop	dictionary ftp-write guest imap named phf sendmail xlock xsnoop	snmp-get
User to Root (U2R)	at eject ffbconfig fdformat ps	loadmodule	perl term	

Surveillance/Probing	ip sweep mscan nmap saint satan	ip sweep mscan nmap saint satan	ip sweep mscan nmap saint satan	ip sweep mscan nmap saint satan

Table 2-1 The DARPA 1998 Attacks, based on (Lippmann, Fried et al. 2000)

Network frames were recorded using TCPDUMP, along with host audit data from the BSM module. Eight different intrusion systems were evaluated as part of the data gathering exercise. All eight performed well when tested using attacks present in the training data. When tested against unseen attacks their performance was poor.

2.6.3.2. The 1999 DARPA Dataset

The 1999 DARPA intrusion system evaluation extended their 1998 work (Lippmann, Haines et al. 2000a). More than 200 instances of 58 different attack types were recorded, with Windows NT hosts now included. Three weeks of training data and two weeks of test data were captured, again with new attacks present in the test data that were not present in the training data. TCPDUMP data was captured both internal to the network as well as external.

Eighteen different intrusion detection systems were evaluated during the trial period. Most systems had false alarm rates below ten per day, but this is usually taken to be a result of the low background traffic. Network-based intrusion system performed well against probe and DoS attacks whilst host-based ones were effective at detecting privilege elevation (R2L and U2R).

The importance of the DARPA datasets to the intrusion system research community cannot be understated. However there has been considerable criticism of their methodology and the resulting data (McHugh 2000). McHugh was concerned that the false alarms generated by the background traffic were not validated experimentally or analytically. He observed that a number of researchers had reported background data types (such as storms of FIN and RST frames) which were not present in the synthetic DARPA data. The limited addition of fragmented frames in the 1999 dataset did not address this issue. The use of ROC curves to determine the best intrusion system was also criticised due to their reliance on the realism of the false alarms.

Also the attacks used within the simulation were representative of the time, which were dominated by Unix with only a few HTTP and Microsoft exploits. Attacks from advanced persistent threats were not included, nor were attacks against peer-to-peer or social media protocols.

Mahoney (Mahoney and Chan 2003) assessed the DARPA data for its applicability to anomaly detection and found simulation artefacts affected the evaluation. They proposed the mixing of the DARPA data with real network data to overcome this limitation.

2.6.3.3. The KDD '99 Dataset

The 1998 DARPA dataset was cleaned for the KDD '99 Cup, a competition associated with the Knowledge Discovery and Data Mining Conference. Connections within the DARPA dataset were used to extract 41 potential intrusion features (Mukkamala, Janoski et al. 2002b). Each connection was

labelled as being from one of five classes namely normal, denial of service, probe, user-to-root and remote-to-local.

2.6.3.4. Other Datasets

In addition to the DARPA and KDD datasets others have been used in intrusion studies. Schonlau (Schonlau, DuMouchel et al. 2001) collected command data from 50 UNIX users, including masqueraders, and made the data available for host-based intrusion research. Greenberg (Greenberg 1988) also collected data from UNIX users.

Moore (Moore and Zuev 2005) have released their hand-labelled dataset that they have been using for the development of Bayesian classifiers. This database is flow orientated and although it has not been collected with intrusion detection in mind it does include attack data.

The Measurements and Operational Analysis Team (MOAT) of the National Laboratory for Applied Network Research (NLANR) maintained a website which contains a large amount of experimental data taken from operational networks in the USA (Wand Network Research Group 2012). This data is larger than the DARPA dataset and more recent. It does not appear to contain any labelling of intrusions but it may become of value in future NIS studies. The NLANR website also hosts the network files taken from the University of Auckland, another valuable resource. In 2006 the NLANR funding was terminated and the Cooperative Association for Internet Data Analysis, CAIDA (The Cooperative Association for Internet Data Analysis 2012) assumed operational stewardship of the NLANR data.

CAIDA have also been collecting and distributing anonymised Internet traffic since 2008, from two high-speed monitors on a commercial backbone link. A considerable amount of TCPDUMP readable frame data is available. They routinely capture an hour of network data every month, at a peak data rate in excess of 8Gbits/second. Whilst this data has been collected using network interface cards with nanosecond precision, there is a significant amount of frame loss. Despite the anonymisation of the data CAIDA place a significant number of restrictions on its use.

The University of Brescia has made available anonymised traces collected on the edge router of their university campus on three consecutive days in 2009 (Brescia 2009). 27GB of TCPDUMP data was recorded containing web (12.5%), mail (0.2%), peer-to-peer (86.1%) and VoIP (1%) protocols. Frame loss was measured as below 1%.

The WAND research group at the University of Waikato Computer Science Department collects and distributes very long trace sets (Group 2013). The Waikato Internet Traffic Storage (WITS) project aims to collect and document all internet traces available at WAND. Over 30 separate trace sets have been captured dating back to July 1999, including wireless network traffic. Fourteen datasets are available for download for networking research.

The WITS data is also available at the RIPE Data Repository. This is a very large data store consisting of approximately 100TB of data, which also hosts some of the NLANR datasets. As well as passive network traces this database includes routing, TRACEROUTE and PING, as well as IPv6 data.

2.6.4. Network Traffic Generation

As an alternative to using recorded network traffic synthetic traffic generators can be used to control the parameters of an evaluation. Specifically, they remove the doubt about whether or not the alert is a false positive or the missed alert is a false negative, allowing repeatable experiments to be undertaken.

HARPOON was an attempt to generate flow-level network traffic for this purpose (sommers and Barford 2004; Sommers, Kim et al. 2004). BRUTE was another software approach to network generation (Bonelli, Giordano et al. 2005), producing IPv4 and IPv6 frames.

Botta et al have reviewed the limitation of modern software network traffic generators (Botta, Dainotti et al. 2010). More recently they have compared software approaches for realistic network workload simulation, proposing a new tool to meet the requirements (Botta, Dainotti et al. 2012). This new tool is based on the D-ITG tool (Avallone, Emma et al. 2004).

In an attempt to generate network traffic at 10GBits/s traffic Bonelli studied the use of multi-core processors (Bonelli, Di Pietro et al. 2012). Non-software based solutions for network traffic generation have also been proposed. Tockhorn et al have developed an FPGA-based generator with the aim to achieve Gigabit link performance (Tockhorn, Danielis et al. 2011).

2.6.5. Comparison Studies

Comparison studies have been limited mainly due to the difficulties of making direct comparison of disparate systems. The DARPA evaluations and the KDD

'99 competition remain the most comprehensive undertaken to date. In addition a number of researchers have used their resulting databases to publish further performance figures.

Mukkamala (Mukkamala and Sung 2003b) compared linear genetic programs (LGP), ANNs, multivariate adaptive regression splines (MARS) and SVMs using five classes and the 1998 DARPA dataset. LGPs were a clear winner in terms of detection performance at the expense of processing time. The performance of the SVMs exceeded that of ANNs and the MARS implementations, and was close to the performance of the LGPs. In previous work Mukkamala and Sung had compared SVMs and ANNs, also concluding the SVMs provide the better detection performance.

2.7. Conclusions

There has been considerable research into the design and operation of intrusion detection systems since the original paper by Anderson in 1980. Much of the published work has focussed on data processing techniques with a wide range of algorithms from pattern recognition and other disciplines applied to this problem, in both isolation and combination. The focus on data processing algorithms has been driven by the perceived shortfall in detection performance, most notably the unacceptably large number of false alarms that can occur.

Strategies for post processing of intrusion alerts, through event correlation or visualisation techniques have attempted to overcome the inherent limitations of intrusion systems and address the increasing workload they can place on system administration staff.

In parallel with the investigation of data processing algorithms there has been significant research into quantifying performance of intrusion detection systems through defining performance metrics, standard intrusion databases and synthetic data generators. Most notable in this was the work of the MIT Lincoln Labs under DARPA funding. This work was initiated over 14 years ago, and many research and commercial intrusion systems have been evaluated, but there still remains no absolute measure of the performance of real systems. Potential users of intrusion systems cannot get definitive statements on the performance of different systems and researchers are unable to say by how much their latest approach is better or worse. There has been no further attempt to extend on the Lincoln Labs work and make it more relevant to current security research or to the needs of intrusion users.

In addition to data processing research, system-level intrusion approaches have also been attempted. In host intrusion systems these have been particularly successful, most notably with anti-virus software. However in network intrusion system applications, whilst they have been successful at reducing false alarms this has been at the expense of unsatisfactory or limiting assumptions.

Despite this extensive body of research into both techniques and systems, there is still no solution to the general problem of confident detection of intrusions with an arbitrary low false alarm rate. It is the assertion of this thesis that one reason for this is the lack of a clear way to compare different approaches from different research groups, in an easily quantifiable way. This inability to quantify and compare systems was most recently observed by Salah (Salah, Macia-

Fernandez et al. 2013) and is a fundamental problem which is addressed directly in this thesis, by proposing a taxonomy designed specifically for comparing systems along with new performance metrics.

In the next chapter the definition of intrusion detection will be examined more closely, to address this difficulty in comparing intrusion systems directly, from which a new taxonomy will be derived.

CHAPTER 3

A NEW TAXONOMY FOR INTRUSION SYSTEMS

3. A New Taxonomy for Intrusion Systems

The literature review described in the previous chapter has identified the difficulty in comparing alternative approaches to declaring the presence of intrusions. This chapter describes a new taxonomy that aims to improve the comparison of intrusion systems (Tucker, Furnell et al. 2006; Tucker, Furnell et al. 2007). Taxonomies are an important aspect of the analysis of systems as they can act as a formal description, providing order to the subject. More importantly, they can provide insights through the identification of gaps. Such insights often identify new areas of research and this was the motivation for developing the taxonomy described in this chapter.

The proposed taxonomy considers the different type of outputs that can be produced by intrusion systems, along with the type of information used to determine the intrusion, as the basis for their comparison. A graphical combination of these parameters is described against which intrusion systems can be qualitatively and quantitatively compared.

3.1. Background

A number of taxonomies for intrusion detection have already been proposed. One of the earliest was undertaken by Debar and classified intrusion systems according to their detection method, behaviour on detection, audit source location, or usage frequency (Debar, Dacier et al. 1999). This was later extended to include the detection paradigm, as either state- or transition-based, where the state of the network or host was determined by the intrusion system (Debar, Dacier et al. 2000). Axelsson offered an alternative taxonomy in terms

of the detection principle and operational aspects, such as whether operation is continuous or in batch mode (Axelsson 2000a). Other taxonomies have been developed that classify intrusion systems according to the attack stage they can declare intrusions, such as pre-attack, real-time or post attack (Lukatsky 2002). Most recently Liao et al have undertaken a comprehensive review of intrusion detection (Liao, Lin et al. 2013) proposing a new taxonomy covering host, network, wireless and behaviour-based intrusion systems. Patel et al have also provided a systematic review of intrusion system applied to cloud computing (Patel, Taghavi et al. 2013), as have Modi et al (Modi, Patel et al. 2013).

Each of these taxonomies provides insight into the operation of intrusion systems and is a useful framework for identifying new research opportunities. However, they are not a good basis for intrusion system comparison as they use the internal properties of such systems for classification. A taxonomy based on the applicability of intrusion systems is a more fundamental comparison approach as it describes their use, rather than the details of their implementation.

Consider, for example, two network intrusion systems. System A is misuse-based whilst System B is anomaly-based. During a series of intrusion events both these systems will indicate the intrusion state of the network segment they are monitoring, possibly to different degrees of accuracy (that is, their detection and false alarm rates may differ). Much work has been published on the quantitative comparison of such systems (for example (Abouzakhar and Manson 2004)), using analysis techniques such as ROC or lift curves. However,

in addition to the presence of an intrusion, System A can indicate the type of attack and the exploit being used, on the basis of the specific signatures that are triggered. Comparing System A with System B via a ROC curve or confusion matrix will not include this important property of System A and thus is not a fair comparison method.

Consider also the use of the output from these two systems. If both systems are providing alerts to network support staff, the actions that are likely to be taken are different. System A will identify the network peers involved in the suspected intrusion behaviour as well as the nature of the attack, allowing support staff to take specific action quickly. Support staff using System B may need to undertake further investigations before the information necessary to stop the intrusion behaviour is derived. In summary, each of these approaches to intrusion detection makes differing demands on the systems that use the information they provide and therefore comparison techniques should include this in their assessment.

3.2. A New Intrusion Taxonomy

A new taxonomy was developed which was inspired by the work of Johnson in the interpretation of pictures (Johnson 1958). He studied the ability of human operators to find and correctly classify objects within complex images. The objects were relatively small and thus the a priori probability that a specific area of the image contained an object was very low, a situation analogous to intrusion events within a background of normal network or host activity

(Axelsson 1999a). Johnson defined the following types of operator tasks, amongst others:

- Detection – the ability to say that something of interest is present in an image;
- Recognition – the ability to determine the class of object present, such as a car or aircraft; and
- Identification – the ability to determine the type of object present, such as the make of car or the type of aircraft.

The most important aspect of Johnson's work was the definition of minimum criteria necessary for successful completion of the above tasks. Using similar task definitions as a starting point, the new taxonomy divides the output of intrusion systems into one of five categories, with the first three loosely in line with Johnson, as follows:

- Detection – in which the system outputs an indication of a state change within a network or host. There is no determination of the nature of the change, apart for the assumption that this indicates the occurrence of a possible intrusion. The principal use of such systems is for data rate reduction so that other systems (either automated or manual) can investigate further;
- Recognition – in which the intrusion systems are capable of declaring the type of attack, such as Distributed Denial of Service (DDoS), reconnaissance, or User to Root (U2R);

- Identification – in which the system is capable of declaring the exploits used to achieve the intrusion, such as buffer overflow or an application-specific vulnerability;
- Confirmation – in which the attack plan is deduced, allowing attack-specific countermeasures to be deployed rather than coarse measures, such as disconnection of the internet access or isolation of key business servers; and
- Prosecution – in which evidential quality data is generated identifying the originator of the intrusion.

This hierarchy is shown in Figure 3-1.

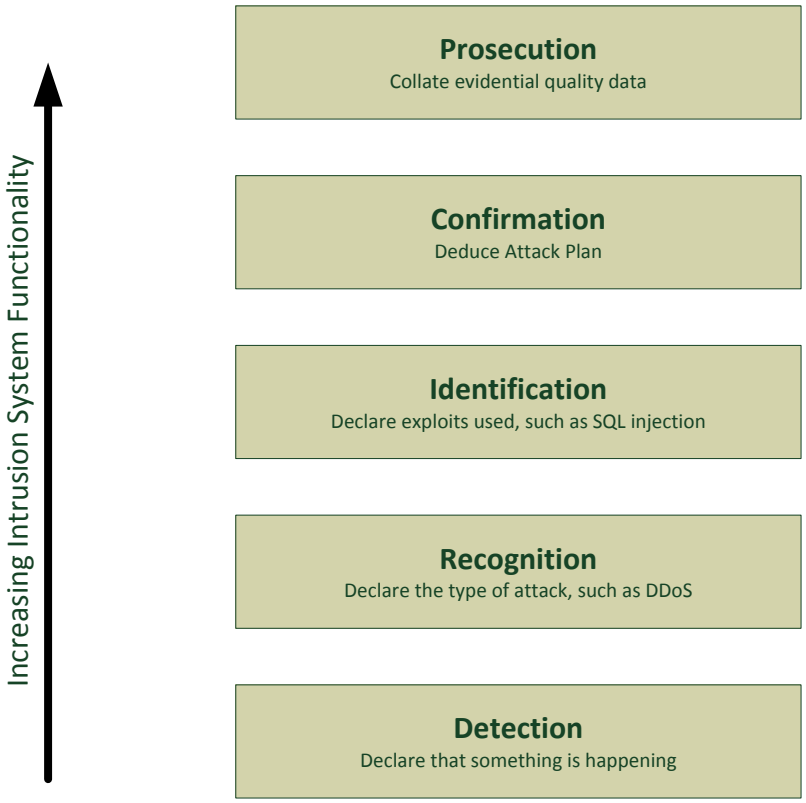


Figure 3-1 Intrusion System Hierarchy

As an example of the use of this taxonomy consider a simple anomaly intrusion

system comparing the utilised network bandwidth with historical values. An alert could be issued when the measured utilisation exceeded the historical levels by a set threshold. Such a system would be categorised as an intrusion "detection" system. It would be able to declare that something unusual is happening within the network but declaring with any confidence that the anomaly was caused by an intruder is not likely to be achievable to an arbitrarily high accuracy.

As another example consider a SNORT intrusion system operating on a single network segment (Roesch 1999). When a rule is triggered and an alert declared, there is considerable attack-related information available. Often, rules are created to alert when the signatures of specific attacks are present. Thus, when such a rule has been triggered, the intrusion system can identify the exploit being used, as well as the network peers involved. Within SNORT, single frames can trigger multiple signatures, allowing detailed attack information to be accumulated downstream of the intrusion system. Within the taxonomy proposed here SNORT is acting as an intrusion "identification" system.

In addition to considering the output from an intrusion system, further insight can be achieved from an analysis of the data scale over which the system is operating. In this context the data scale means whether or not the data is local to the host or application, or more widespread data, such as local area network frames are available. In modern computer systems four data scales can be considered:

- File – monitoring the status of individual files for unauthorised access or change;
- Host – monitoring the applications running on and the behaviour of an individual host or user;
- Network – monitoring the frames exchanged between hosts, servers and other network devices to assert the presence of an intrusion; and
- Enterprise – monitoring traffic originating from trusted sources of an organisation that operate in the presence of other, less trusted, data sources.

The File, Host and Network data scales have been used in other studies (for example, (Bace and Mell 2001)). However, the separation of the Network data scale into two sections is believed to be a novel concept. The principal difference between the Network and Enterprise data scales is the mixing of trusted and untrusted data streams within the same network segment. This is most often encountered in virtual private networks (VPN) between an office location of an organisation and its remote staff or trusted partners, via the Internet. VPNs are separated from the untrusted data streams using encryption schemes and well-known protocols. However, this separation may become subject to the same technology, policy or configuration vulnerabilities as other parts of the information processing system. Therefore, it is likely that security staff will want to know when their VPN communications are subject to intrusion attempts, irrespective of whether or not the attempts are successful. Intrusion systems therefore need to extend their data scale applicability to include the Enterprise. This is a technically challenging problem.

There is an alternative to the File and Host scales that could be applied in specific analyses, if needed. These are Application and Operating System. It is becoming increasingly common for enterprise applications to include data gathering for intrusion declaration sub-systems, independent of other security features of their host. Such applications are only concerned with intruder behaviours associated with their own files and are therefore more limited that the File scale used previously. This limitation is countered by the Operating System scale, which addresses more general issues of file-based intruder activity. In essence this alternative scale moves the separation of the lower two scales nearer to the first. For the remainder of this thesis the use of the File and Host scales will be used to describe this taxonomy.

3.3. The Application of the Taxonomy

This taxonomy can be applied in a number of ways. The remainder of this chapter will examine its use to create an intrusion footprint on a grid or matrix formed from the output type and data scale elements of the taxonomy. The use of this footprint for comparison of systems will then be shown.

3.3.1. Intrusion Matrix

The combination of intrusion output type and data scale can be shown as an intrusion matrix, as in Figure 3-2. Also shown are some of the techniques that can be applied within a particular output type and data scale. For example, malware signatures or resource anomalies can be used in intrusion recognition systems operating at the Host data scale. Much of this matrix is covered with techniques that have been extensively studied. Of particular note is the

difficulty of practical techniques at the Enterprise data scale, when applied outside a managed cloud.

	FILES	HOST	NETWORK	ENTERPRISE
PROSECUTION	Protective Monitoring, Secure Data Vaults	Protective Monitoring, Secure Data Vaults	Ant-spoofing, Trust Management	Cloud-based Security Services
CONFIRMATION	Attack Trees	Attack Trees	AI Techniques	Cloud-based Security Services
IDENTIFICATION	Malware Signatures	Malware Signatures	Exploit Signatures	Cloud-based Security Services
RECOGNITION	File Hashes	Malware Signatures, Resource Anomalies	Traffic, Protocol or User Anomalies	Cloud-based Security Services
DETECTION	File Hashes	Sys Calls, Registry Use, Resource Anomalies	Traffic, Protocol or User Anomalies	Cloud-based Security Services

Figure 3-2 Intrusion Taxonomy

3.3.2. Intrusion System Footprint

The intrusion matrix can be used to plot a footprint for different intrusion systems. The footprints are determined from an analysis of the intrusion system outputs to determine which of the five output categories the system is capable of producing and what data scale is used to create the output. For example, Figure 3-3 shows the footprints of a number of different intrusion paradigms.

Figure 3-3a shows the footprint of representative anti-virus software (AVS) package. They typically include both virus-specific signatures and heuristics that respond to anomalous behaviours. This means they operate from the Detection to the Identification output types. Since the attack plan can often be

determined by reverse engineering of the virus, AVS packages can also be considered to operate at the Confirmation output type.

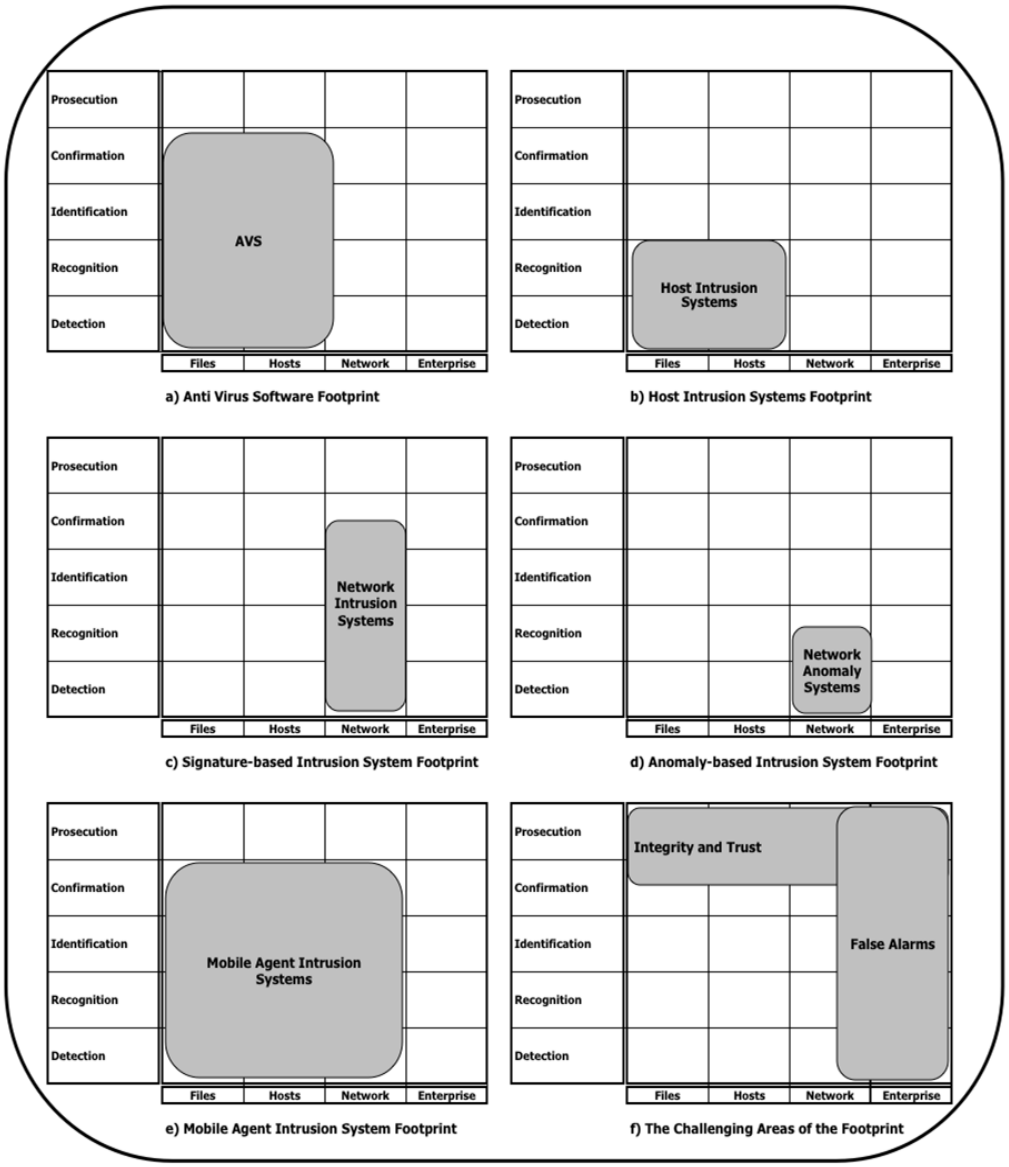


Figure 3-3 Intrusion Footprints

A footprint of a host-based intrusion system is shown in Figure 3-3b. To create this footprint it was assumed that anomaly techniques are applied and therefore the intrusion system is only capable of Detection or Recognition. Confirmation, or the determination of the specific exploit or vulnerability used (Identification), are unlikely to be achievable with confidence when using an anomaly-based

system. Host-based intrusion systems using signature techniques would be expected to operate at the Identification and Confirmation levels, depending on the discrimination capabilities of the signatures.

Figures 3-3c and 3-3d show network-based intrusion systems using signature and anomaly detection respectively. These figures highlight the principal differences to be at the higher output types of Identification and Confirmation. SNORT is a typical example of a signature based intrusion system. On its own it is unable to perform the plan determination required for full Confirmation. However, when multiple SNORT sensors are deployed at strategic parts of a network, it may be possible to determine an attack plan from the patterns of signatures that are triggered. An additional module would be required to integrate the information and determine the plan. Hence, the Confirmation output type is shown partially covered by the footprint.

Figure 3-3e is the most interesting, and shows the extensive footprint that could be achieved by intrusion systems based on mobile agents (see section 2.3.3 for a review of mobile agents applied to intrusion detection). On the assumption that mobile agents could be created to examine the status of files, applications running on a host, and frames on local network segments, they offer the widest range of data scales of any other technique. Also, their payload could include integrated anomaly and signature-based techniques, and when combined with a communications capability this could give them the potential to provide output types up to Confirmation. It may even be possible that

techniques for Enterprise data scales and Prosecution could be integrated as they become available.

Finally, Figure 3-3f shows some of the current challenges faced by intrusion systems. The Prosecution output type requires high integrity information to be gathered and secured from change. Whilst this is a common requirement in secure systems it must be achieved to the levels necessary to allow criminal prosecution, within a system that has intruders present (Sommer 1999). For the Enterprise data scale, the technology challenge appears to be the development of discriminants that will separate intrusion and non-intrusion events in mixed-trust data flows. Such data flows will often be occurring on equipment not owned by the enterprise and therefore the ability to provide local monitoring of the network will be limited.

It is useful to consider how Security Information and Event Management (SIEM) fit within this footprint. An SIEM can aggregate data from all the data scales at all intrusion levels, assuming the Enterprise level is restricted to procured cloud services only. This would mean that they could cover the intrusion footprint completely, providing all security information is sent to the SIEM. However an SIEM does not undertake the measurements on the system directly, but uses measurements made by other system elements, such as applications, operating systems, network devices and intrusion systems. Although the taxonomy described here, as well as the metrics described in the next chapter, could be applied to SIEM, this lack of inherent measurement means that they will not be considered further.

3.3.3. Comparison of Intrusion Systems

The intrusion matrix can be used to provide a comparison between systems. A qualitative comparison can be made by examining the footprint of each system. Large footprints are likely to represent systems that provide a broader range of applicability and a wider range of output information during an intrusion. Small footprints would be typical for systems that are very specific in their application.

A more quantitative comparison can be made by examining the performance of systems where their footprints overlap. Each element of the intrusion matrix is accompanied by a set of performance metrics relevant to the output data type. These performance metrics could include false alarm rates, intrusion probabilities, or confusion matrices measured in such a way as to be appropriate to the position within the intrusion matrix. As an example consider a single element within the intrusion matrix, say the (Network, Identification) element. If the footprints of two intrusion systems overlap on this element, performance metrics relevant to Identification should be calculated for the two systems. The probability of identification could be determined as a function of the false alarm rate, to produce Identification ROC curves. Examination of the ROC curves at this overlap point within the intrusion matrix would allow comparison of the systems in the role of intrusion identification. A fair comparison would require the examination of performance metrics at all points of overlap on the intrusion matrix as well as a recognition of the additional capabilities offered at points where they do not overlap.

Some of the elements of the intrusion matrix presented have been extensively

studied and can be considered commercial successes. For example, AVS packages are very good at providing confident alerts at the Files and Host data scales (Post and Kagan 1998). Such software can be very specific, identifying the virus and hence, by implication the "plan" of the originator of the virus. Heuristic algorithms can provide a degree of detection capability in which the AVS indicates that there is a virus present but is not specific about its type. AVS packages are also well known to provide a high alert probability with a low false alarm rate. Thus a large area of this matrix can be achieved with very high performance.

Meanwhile, some of the elements of the intrusion matrix are poorly understood at this time. Effective techniques at the Enterprise data scale are rare and of limited applicability. This applies to any of the intrusion output capabilities. The enterprise data stream may be present with untrusted streams and on untrusted network equipment (for example, Internet backbone routers). Current intrusion systems are not able to operate outside of the trusted systems of the enterprise, except in limited circumstances, leaving Enterprise scale intrusion systems to rely on remote diagnosis of intrusion behaviour. However some aspects of the trust issues are addressed in cloud computing where the provider of the cloud can be subject to contractual and service level agreements for security, in which intrusion declaration could form a part. In this context the cloud is equivalent to leased infrastructure, rather than the general purpose Internet infrastructure that would be difficult for a single organisation to monitor.

It can therefore be seen that there are three aspects of the intrusion matrix that provide insight to the performance of an intrusion system and should be considered when comparing systems, namely:

- The number of elements of the matrix that an individual system footprint covers as this can indicate its applicability;
- The position of the elements of the footprint within the intrusion matrix, as some element positions present a significant challenge to the achievement of high performance; and
- Only the elements that overlap are of any significance in the direct quantitative comparison of intrusion systems.

3.4. Relationship with Other Definitions of Intrusion

One of the earliest definitions of intrusion was from Amoroso. He defined intrusion detection as "*the process of identifying and responding to malicious activity targeted at computing and networking resources*" (Amoroso 1998). In the same year Ptacek defined intrusion as "*unauthorized usage of or misuse of a computer system*" (Ptacek and Newsham 1998) whilst Alessandri defined intrusion as "*a malicious activity threatening the security policy that leads to a security failure, that is to a security policy violation*" (Alessandri, Cachin et al. 2001). More recently many researchers have used the definition of Bace in which intrusion is defined as "*attempts to compromise the confidentiality, integrity, availability, or to bypass the security mechanisms of a computer or network*" (Bace and Mell 2001).

For the remainder of this thesis we will use a simple definition based on Alessandri, without the restriction of malicious intent. Therefore we consider intrusion to be defined as "an activity that leads to the violation of the security policy of a computer system". Further insight can be gained by considering the relationship of this definition with the definitions of detection, recognition and identification presented earlier.

In this context, an intrusion detection can be seen as the declaration that the security policy has been violated, but the specific clause that has been violated is not identified. Intrusion recognition systems are able to declare which clauses or subsets of clauses have been violated. Intrusion identification systems are able to declare which clauses or subsets have been violated, as well as declaring the way in which they have been violated.

The above discussion can be used as the basis of a mathematical model of the intrusion declaration process, potentially allowing the theoretical limits to be determined in the same manner as Johnson's work for imaging systems. Also the inclusion of AVS within this taxonomy opens the challenging and interesting option of building on the theoretical work already published in this area. The work of Cohen (Cohen 1987) has already established theoretical limits on the detectability of viruses, proving that no algorithm can perfectly detect all possible viruses. More recently Li et al have proposed a theoretical basis for intrusion, but this work has yet to reveal any useful conclusions (Li, Das et al. 2005). It is hoped that this taxonomy will build on this theoretical basis and

lead to a better understanding of the limits of performance for intrusion systems, as well as providing an improved framework for their comparison.

3.5. Conclusions

Published research literature in intrusion detection has failed to define precisely and consistently the meaning of "detection" and therefore comparison of detection systems is problematic. Indeed, it is common for some researchers to refer to intrusion detection, whilst others to intrusion recognition within the same context.

In this chapter a new taxonomy for intrusion systems has been defined in terms of five levels of intrusion operating over four data scales, producing an intrusion system footprint. The footprints for different types of intrusion system have been examined with the qualitative observation that the larger the footprint of a system the larger its applicability to intrusion problems. Intrusion systems can only be compared quantitatively in a meaningful way when they have overlapping areas on their footprint.

Existing definitions of intrusion detection have been examined. The relationship between the new levels of intrusion functionality has been defined in terms of breaches of the security policy of a system with intrusion "detection" being defined as "an activity that leads to the violation of the security policy of a computer system". This chapter has shown that intrusion systems can only be meaningfully compared when they are attempting to do the same task, over the same scale.

The following chapters will concentrate on the detection, recognition and

identification scales only with the next chapter providing a systems-level discussion of intrusion, focusing on the Network scale.

CHAPTER 4

SYSTEMS CONSIDERATIONS

4. Systems Considerations

The previous chapter has established an alternative view of intrusion systems, in terms of the different types of outputs that can be produced. This chapter examines the systems implications of these different output types. Although many of these implications can apply across different scales, for this chapter and for the remainder of this thesis the emphasis will be on network intrusion systems (NIS) and, in particular, signature-based NIS. The acronym NIS is proposed to mean network intrusion systems operating at the detection, recognition and identification scales, to avoid confusion with the more widely used NIDS (network intrusion detection system) acronym. It is introduced to clarify that the systems considerations apply more broadly than to detection alone.

The chapter commences by establishing a common terminology and understanding about the nature of signature-based NIS. This is followed by a discussion of the reasons for deployment and the characteristics of an ideal NIS. These characteristics are illustrated in terms of a model of an ideal NIS and a discussion of the current challenges facing NIS. Finally, two performance metrics are proposed to assist in the use of the new taxonomy.

4.1. Principles of Network Intrusion Systems

Currently there is no complete theoretical treatment of NIS. There has been considerable modelling of network traffic (Barford, Kline et al. 2002b; Allen and Marin 2003; Estevez-Tapiador, Garcia-Teodoro et al. 2003; Jun, Jiahai et al. 2005; D'Apice, Khokhlov et al. 2010; Bahaa-Eldin 2011) and attempts to

establish a full theoretical model (Patcha and Park 2004; Li, Das et al. 2005; Beghdad 2009). However it is difficult to use these results to further the design of better NIS. In this section some basic theoretical results will be stated as a pre-cursor to establishing performance metrics later in the chapter.

In order to establish the important issues associated with a signature-based NIS it is useful to view it from a set theory perspective of intrusion alerting, as follows. Signature-based NIS generally operate by comparing each network frame individually against a set of frame-based signatures that represent malicious behaviour of interest. This is a simplified view as intrusion systems can contain pre- and post-processors which can extend their operation beyond a single frame. The frag3 pre-processor in SNORT is an example, re-assembling multiple fragments of a frame, therefore extending the application of the signatures over several network frames. The stream4 pre-processor, also in SNORT, gives it the capability to alert on frames according to where they are in a connection, that is, according to connection state. Post-processors can be used to limit the output from an intrusion system, suppressing previously alerted conditions for example, extending the decision process of the NIS over many frames or connections.

Although this single frame view is a simplification it does offer some insight into NIS operation. Consider intrusion alerting from the perspective of the universe of network frames, U, collectable on a network segment. This is a very large set. For a maximum transmission unit (MTU) of 1500 bytes, the number of potential frames at the full MTU is 2^{1500x8} which is approximately 10^{3612} or

considerably larger than the baryon number of the observable universe (~10^{80}). Included in this set are frames that meet the defined network protocols and their options for payloads, as well as those that violate all current protocols. Despite the large number of set members some simple results can be derived.

There exists a subset of U that consists of the frames that may be used by intruders to violate the security policy of a network. These could include, for example, frames in which there are protocol violations, to defeat the TCP/IP stack of the target machine, or frames that comply with the network protocols but deliver malicious payloads. This subset is designated, S_i, is considered to be a proper subset to ensure that there is the possibility of separating intrusion-like from non-intrusion-like frames, that is

$$\mathbf{S_i \subset U}$$

Equation 4-1 Fundamental Assumption of NIS

Equation 4-1 illustrates a fundamental assumption of signature-based network intrusion, that S_i is a proper subset of U. This means that there are some frames, and hopefully many, that can only be the result of behaviours consistent with the network security policy and therefore can be discriminated by a NIS on a per-frame assessment. An important consideration is that S_i is not dependent on the specific implementation of a signature-based NIS, but is a fundamental property of a network and specifically its security policy, network architecture, protocols and vulnerabilities. The size of S_i is not fixed as the discovery of new vulnerabilities and attack methods, or the deployment of devices using new protocols, will affect the number of frames that comprise this set.

In the universe of frames there also exists a subset that consists of frames that can be generated as a result of actions consistent with the network security policy. This subset, designated S_n is not necessarily a proper subset, hence it can be defined as $S_n \subseteq U$. It is also independent of the design of a signature-based NIS.

A problem faced by signature-based NIS is that:

$$S_n \cap S_i \neq 0$$

Equation 4-2 The Problem of NIS

This is a fundamental limitation of signature-based NIS. It is not possible to uniquely map each network frame into either S_n or S_i and therefore error free classification of all frames as either intrusion-like or non-intrusion-like is not possible. Some frames are consistent with normal user and intruder behaviour simultaneously. Therefore perfect declaration of an intrusion cannot occur by considering only single frames and matching them to known intrusion signatures.

Equation 4-2 can be proven by construction. Consider for example the use of ICMP PING. Some applications use PING to verify connectivity to remote servers. However an intruder can also use PING to locate potential hosts to attack. The presence of a PING frame on a network segment therefore does not necessarily indicate that an intruder is present yet it is common for alerting on PING frames to be included in a signature set for a NIS.

No published research could be identified that has quantified the overlap between S_n and S_i. Practical experience suggests that the overlap could be

large, mainly due to the frames present in techniques used legitimately by support staff, which would be considered malicious when used by unauthorised individuals.

Although S_n is not dependent on the signatures used within a NIS, it is highly dependent on the network security policy. Re-considering the PING argument given above highlights this relationship. If the network security policy forbids the use of PING by users and support staff and also forbids the deployment of applications or operating systems that use it autonomously, then alerting on a PING on a network is a valid intrusion declaration as the policy has been violated. The PING frames would exist only in the set S_i and not in S_n.

Forbidding PING is not a practical solution to the limitations of signature-based NIS. However it demonstrates an important point that making the network security policy more specific reduces the size of S_n and hence the overlap with S_i.

Consider next the subset of U that comprises the frames on which a given set of NIS signatures will produce an alert, designated S_{sig}. For the perfect NIS the following equations hold true:

$$S_{sig} = S_i$$

$$S_{sig} \cap S_n = 0$$

Equation 4-3 An Ideal NIS

This is unlikely to be achievable for practical systems due to the implications of the limitations highlighted in Equation 4.2. Also the size of S_i is unknown, due

to as yet undiscovered exploit methods. In essence the first part of Equation 4-3 is concerned with the achieving high detection probability and low false negative rates, whilst the second part is concerned with achieving low false positive rates.

In this simplified situation of decisions made on single frames, the goal of the misuse NIS designer would be to select signatures to get as close to this ideal relationship as possible. This could be done in two distinct ways:

- Increasing the number of exploit-specific signatures; and
- Decreasing the specificity of the signatures.

Increasing the number of signatures could provide increasing coverage for new attack methods as they are discovered. Highly specific signatures are less likely to be triggered by legitimate user actions. However as the number of signatures is increased there would be practical difficulties in applying them to network frames in real-time, particularly as network speeds increase.

Decreasing the specificity of signatures would increase the coverage of S_i even for as yet undiscovered attacks, but increase the likelihood that frames that are not intrusion-like would cause alerts, that is $S_{sig} \cap S_n$ would increase.

4.2. Reasons to Deploy a Network Intrusion System

It is instructive to consider the reasons for the deployment of a NIS, to assist with the development of a better understanding of false alarms. A NIS could be deployed for a number of reasons, including to:

- Comply with industry standards – Standards, such as the Payment Card Industry Data Security Standard (PCI-DSS) (PCI Security Standards

Council 2010), often require that intrusion systems are deployed. Compliance is often mandatory and confirmed by independent audit;

- Demonstrate secure operation and corporate governance – Organisations teaming with industrial partners or seeking business with new clients are often required to demonstrate that information assurance processes are effective. Pre-bid questionnaires are frequently used by Government and large organisations to screen potential bidders for major contracts. If the contract involves access to sensitive information the questionnaire is likely to request network security information as well as security standards compliance;
- Gather forensic information to enable a criminal prosecution – A NIS in which the logs are correctly managed could be used to initiate disciplinary procedures or criminal action against employees, or others accessing or using data assets inappropriately and against the network security policy;
- Block an on-going attack, limiting further compromise – Determination of an intrusion in real-time, or early in the attack, can enable action to be taken to limit compromise of systems or information assets. Intrusion Prevention Systems (IPS) can drop frames, slow connections or terminate sessions to deter, limit or stop the actions of an attacker (Papadaki 2004);
- Undertake post-intrusion damage assessment – After an attack it can be essential to understand what information assets or systems have been compromised, for example for compliance with regulations governing

management of personal data. The owners of the information assets may need to be informed, so that they can take appropriate action. A NIS is potentially one source of data, along with server, firewall and other network device logs; and

- Discover attack methods – An NIS is just one component of a defence in depth strategy for network security. As new attack methods are developed it is essential that the network remains secure. Analysis of attacks alerted by a NIS can be used to confirm that the security controls in the other layers of the defence remain effective.

4.3. The Ideal Network Intrusion System

Analysing the concept of an ideal NIS is a useful way to understand the limitations of current systems. An early attempt to document the properties of an ideal NIS was undertaken by Cramer, who identified the characteristics as timeliness of response; high probability of detection; low false alarm rate; specificity of attack; scalability to large networks and low a priori information requirements (Cramer 1995). Later, Lin was concerned about implementation issues (Lin, Tseng et al. 2001) and defined the properties of an ideal intrusion detection system as "*an efficient detection mechanism and provide good representation of expert knowledge for intrusion patterns, which should be easily understood and maintained*". Behera extended Cramer's list to include the limited use of host and network system resources, flexibility in detecting new attacks and the ability to correlate data from different machines to detect coordinated attacks (Behera 2001). Chinchani additionally recognised as important the ability to be deployed in a heterogeneous and distributed

environment (Chinchani, Upadhyaya et al. 2002). The most recent set of ideal characteristics is on the COAST website which provides the following list (COAST 2012):

"*an ideal NIDS should run continuously; be fault tolerant; be resistant to subversion; have minimal overhead on the network and hosts; observe deviations from normal behaviour; be easily tailored to the system being monitored; be able to cope with changing system behaviour; and must be difficult to fool*".

In this section a more general view of the set of ideal characteristics for a NIS is taken, derived from the viewpoint of the individuals responsible for network management. The properties identified above fail to capture all of the desirable characteristics of an ideal NIS from this viewpoint and do not allow their inter-relationships to be clearly seen. A recent information security breaches survey (PricewaterhouseCoopers 2012), indicates a shift from information security being an expenditure, towards being an investment. In the UK 20% of companies require a return on investment (ROI) calculation to support expenditure on information security. The ROI calculation would need to take into account, for example, the financial impact of reputational damage or loss of new product design details, if the information assets of the organisation were to be compromised. To provide such detailed justification, network managers require a clear understanding of the benefits of NIS, the achievable performance and the total cost. Therefore, from the perspective of individuals responsible for managing computer networks, an ideal NIS should:

- Improve the security of the network;
- Achieve high sensitivity, in which high probability of intrusion alerts are

achieved with low false alarm rates;

- Achieve high selectivity, in which the different intrusion mechanisms can be differentiated so as to allow a response specific to the intrusion;
- Be appropriate to meet the threat; and
- Have low cost of ownership.

It may seem unusual to include the requirement to improve the security of the network in the above list, but it is the raison d'être for intrusion systems and therefore it is essential that it be explicitly evaluated. Each new device or protocol stack added to a network has the capability to introduce new vulnerabilities through technology weaknesses, configuration errors, or security policy inadequacies. The net result does not automatically mean that network security is increased, as the work of Ptacek showed (Ptacek and Newsham 1998). Additional security concerns that need to be explicitly evaluated include:

- The ability to operate and survive during a direct attack on the NIS;
- Ensuring that the logs are handled in a forensically secure manner to enable criminal prosecutions to be sought;
- The tolerance to equipment failure is appropriate, with the most secure networks requiring a fail-secure approach;
- Both previously seen and unseen intrusion mechanisms must be detectable; and
- The NIS must not be vulnerable to evasion and insertion attacks, as well as other defeat techniques (Ptacek and Newsham 1998).

The need for high sensitivity encapsulates the requirement for detection, recognition and identification probabilities to be high, simultaneously with an

acceptably low false alarm rate. From a detection theory viewpoint, this requires a large signal to noise ratio, where the signal is considered to be a measurement in which an intrusion is present and the noise is considered to be a measurement in which an intrusion is not present (Schwartz and Shaw 1975). The close relationship between false alarms and the detection of real intrusion events is most often shown by ROC curves, as described in (Abouzakhar and Manson 2004). High sensitivity implies a sharply rising ROC curve achieving high alert probabilities under low false alarm rates. It is important that high sensitivity is maintained across the complete set of intrusion measurements, including unseen attacks, intrusion events during direct attack of the NIS and under conditions of multiple intrusion events from many attackers.

High selectivity is concerned with the ability to discriminate between different intrusion mechanisms and is linked with the need to determine the cause of the intrusion, to be able to select or recommend the most appropriate response. The previous chapter discussed the characteristics of intrusion detection systems with differing intrusion recognition and identification capabilities (Tucker, Furnell et al. 2007). As in the need for high sensitivity, it is important that high selectivity is maintained across the complete set of intrusion measurements, including unseen attacks, intrusion events whilst the NIS is under direct attack and under conditions of multiple intrusion events from many attackers.

The appropriateness of the NIS covers a wide range of properties. It is most directly related to the "timeliness of response" parameter described by Cramer.

However the importance here is that the NIS must be capable of meeting the security requirements stated within the network security policy, rather than be able to respond rapidly. For example, in a school network the Governors may be concerned with the need to protect the integrity of individual student grades. An offline intrusion system analysing log files in batch mode could easily be capable of meeting their security policy requirements. However, in a commercial organisation generating revenue solely from web transactions, real-time response would be essential to block attacker activities before customer details had been compromised and the company reputation permanently damaged.

Other appropriateness-related issues include:

- Network coverage – in which the NIS is required to meet differing threat protection levels in different parts of a network, throughout the enterprise;
- Temporal coverage – in which the NIS is required to meet continuous or batch processing requirements;
- Adaptive performance - based on changing threat, network conditions or availability of resources such as network staff; and
- The ability to gather evidential quality information to assist prosecution of intruders.

The final characteristic of an ideal NIS is low cost of ownership, which generates a number of additional considerations, including:

- Ease of deployment – centralised deployment and management of a NIS will reduce ownership costs significantly;
- Sensitivity to deployment constraints – When the performance of a NIS is sensitive to constraints such as the location of network sensors, the cost of ownership can increase. Space in data-centres is often charged at a premium and therefore NIS that can be deployed flexibly, without loss of performance, will be more attractive from the cost of ownership viewpoint. An additional but important consideration is the constraints placed on other systems that can be deployed after the selection and deployment of a NIS;
- Support requirements – should be minimised by reducing the need for a priori information, routine maintenance including signature updates and specialists to investigate alerts. The achievement of high sensitivity and selectivity are key to reducing the support costs as time will not be wasted investigating false positives;
- Resource usage – increasing the use of network resources, such as bandwidth, storage and processing power, increases the cost of ownership by requiring upgrades to the network infrastructure earlier than would otherwise be needed;
- Scalability – As organisations grow it is important that the NIS solution can scale and that complete replacement is not essential when a threshold network size is reached; and
- Heterogeneous and distributed networks – should not be a constraint. Networks are rarely homogeneous, often containing legacy protocols and

equipment. A NIS that requires significant network upgrades or standardisation is likely to need a large initial investment.

4.4. A Model of an Ideal NIS

Network intrusion systems generally sense the network traffic on key network segments and look for the presence of intrusion signatures or anomalous behaviour. The declaration of an intrusion is then made via the examination of the properties of individual frames, or on the statistical parameters of complete connections. NIS generally do not interact with the network and often their ability to transmit frames onto the monitored network is disabled by hardware, via modifications to the network interface card or cable. This is due to the need to hide the presence of the intrusion system from would-be attackers. Figure 4-1 shows a simple model of a passive NIS.

In this model the Extract Measurements block processes network frames to determine fundamental features of the received data. Such features could include session indicators, frame size, protocol types, or client and server information. The selection of features is one of the most important aspects in the design of NIS. Frequently, more than one measurement is made, allowing an assessment of network activity to be created from many viewpoints (Mukkamala and Sung 2003d). The Extract Measurements block can contain local storage, to allow the extraction of measurement over many network frames. However, when measurements are required over an extended time period it is more likely that partial measurements will be passed to the Associate Measurements block for storage in the Potential Intrusions store.

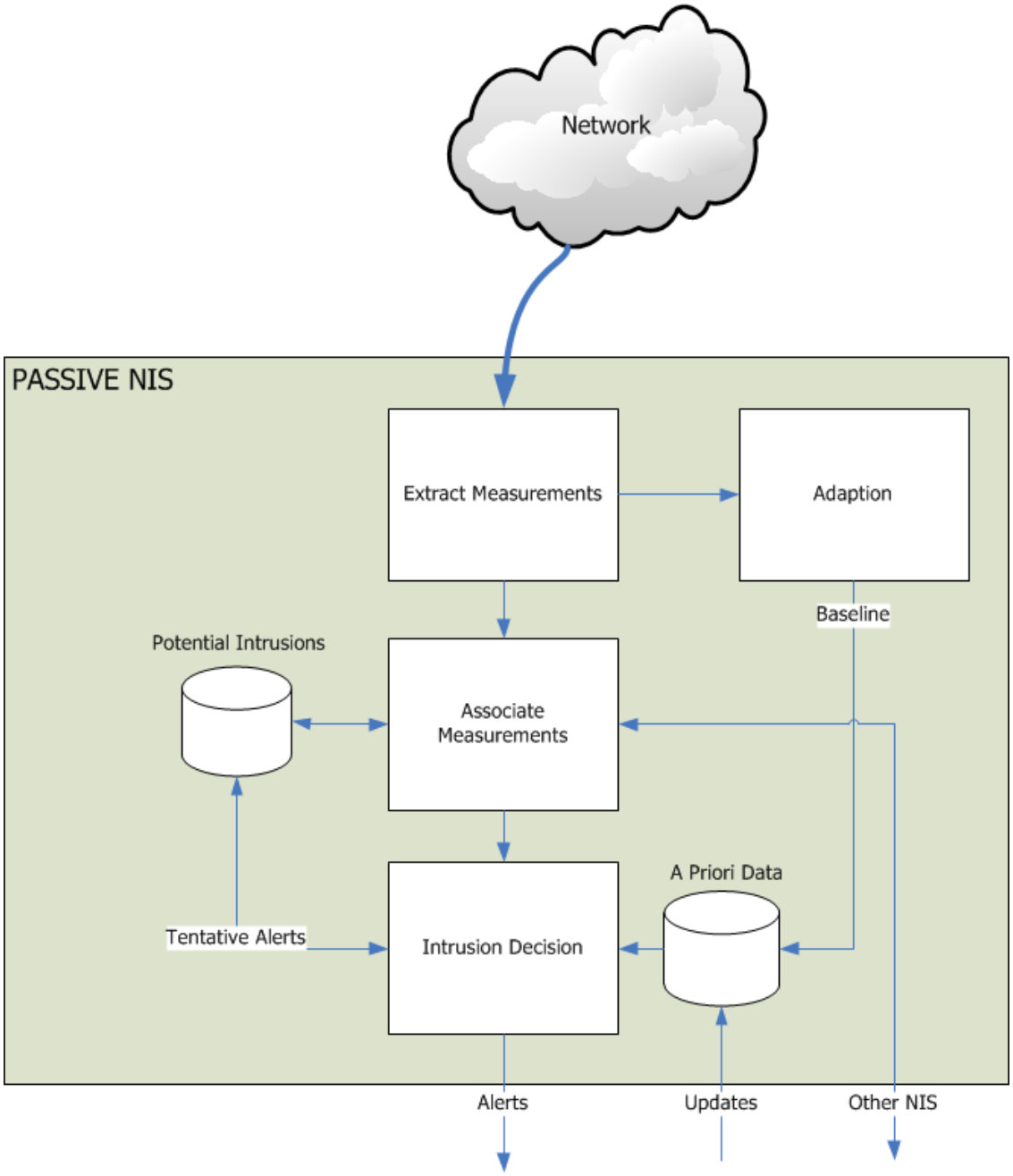


Figure 4-1 A Functional View of an Ideal NIS

The Associate Measurements block takes the latest measurements, partial measurements or inputs from other NIS, and attempts to assign this data to individual records in the Potential Intrusions store. Bass discusses the importance of association techniques in the fusion of multiple sensors (Bass 2000). Each record in this store contains the set of measurements on which an

intrusion/ non-intrusion decision can be made. It is important to realise that these sets will be in error due to uncertainties in the measurement process, incomplete data, an intruder deploying deliberate measures to deceive the NIS or to association errors.

The Intrusion Decision block takes the tentative records from the Potential Intrusion store and, using the supporting information in the A Priori Data store, classifies each record as intrusion-like or non-intrusion-like. It may also separate intrusion events into individual intrusion attack types, to enable the selection of the most appropriate response, reducing the network support requirements. The Potential Intrusion store is updated with the result from the Intrusion Decision block and if an intrusion is declared then an alert is output from the NIS for further action. This block may make its decision in a simple way, for example by checking for the presence of protocol errors, or in a more complex way, such as utilising neural networks or support vector machines to transform the measurements before a threshold or decision measure is applied.

Also shown in Figure 4-1 is an Adaption block taking measurements from the Extract Measurements block and updating the A Prior Data store. This block, if present, can extract information to support models of normal behaviour on which many anomaly-based NIS depend. Often the A Prior Data block is not dynamically updated from network measurements, but is derived from offline analysis of network vulnerabilities or specific attack mechanisms.

Not all of the blocks and stores shown in Figure 4-1 need to be explicitly present. Consider for example SNORT. In its simplest form, that is a single

network sensor with no pre-processors, the Extract Measurements block becomes the capture of a complete network frame. There is no Potential Intrusions store and therefore the Associate Measurements block only passes the frame to the Intrusion Decision block where it is parsed to look for the presence of intrusion signatures taken from the A Prior Data store. The presence of a signature generates an alert.

The discussion above represents a simplified view of network intrusion systems and is not thought to have been presented elsewhere. In this respect it can be considered as a model of a first generation NIS, with modern implementations extending the intrusion assessment over many frames, as described in section 4.1. One motivation for its inclusion here is that other technologies use a similar model, and therefore this can be thought of as a design pattern. The most notable example of its use is in air traffic control (ATC) radar, as described below.

ATC radar makes regular measurement of the position and velocity of aircraft and associates each new set of measurements with the tracks of previously seen aircraft, to update their position and velocity. If the measurements cannot be associated with an existing aircraft then a new aircraft is declared and a track file is initiated. When track files have not been updated for a set period the file is deleted, on the assumption that the aircraft is no longer in the ATC radar control space. In this design pattern, the position and velocity measurements assume the same significance as the measurement of network frames. The Associate Measurements block assigns new position and velocity

measurements with the existing aircraft tracks, in much the same way as it performs the association of measurements in a NIS.

For radar however the problem of how best to associate new measurements with previously seen events has been studied extensively, with techniques based on probabilistic data association, Kalman filtering, interacting multiple models and multiple hypothesis tracking being routinely applied (Kirubarajan and Bar-Shalom 2004). The use of these techniques in NIS is still to be studied in depth, despite their well-known performance in radar.

4.5. Current Challenges in NIS

The passive approach to network intrusion has a number of difficulties in achieving the ideal NIS characteristics described in the section 4.3. Surprisingly, one of the biggest challenges is with the need to improve security. When NIS have a high false alarm rate the network support staff are likely to disbelieve alerts, often taking no action when they occur. Even when low false alarm rates are achieved, low intrusion alerting probabilities can generate a false sense of security in the network support staff. This can occur, for example, when previously unseen intrusions are present and the lack of alerts may result in the support staff believing that the network is secure.

This last point is important and reveals a significant limitation of current NIS. The assumption that a NIS is correctly asserting the status of a network or network device as in one of two states, that is intrusion-free or not intrusion-free, is flawed. The measurements taken by a NIS to make this assertion are

generally not capable of confirming which state a network is actually in, without further assumptions.

For example, consider a simple signature-matching NIS, such as SNORT. A frame is captured from the network (measured) and checked against the known signatures of intrusion-like frames. If the frame is not in the set of known intrusions the NIS does nothing, awaiting the next frame to check. However, if the frame is in the set of known intrusions an alert is issued. The presence of a frame that does not show intrusion characteristics cannot be used to assert that the network is free of intruders. It is supporting evidence for this assertion, but it is not sufficient. An intruder could be using new, previously unseen techniques to compromise the security of the network. The combinations of alert or no alert, along with the presence or absence of evidence of hacking can be seen in Table 4.1. Only one condition is capable of correctly asserting the security status of the network, which is the correct processing of an intrusion-like frame to declare an alert.

		Hacker Signature	
		Absent	**Present**
NIS Output	**No Alert**	**INSUFFICIENT INFORMATION**	**WRONG ASSERTION**
	Alert	**WRONG ASSERTION**	**CORRECT ASSERTION**

Table 4-1 The Assertion Matrix

Consider further the "Correct Assertion" entry in this table. Equation 4-2 shows that even when an intrusion-like frame is present and the NIS outputs an alert, it does not necessarily mean that an intrusion is present in the network.

This limitation reveals more concerns regarding the operation of NIS, when combined with the work of Axelsson (Axelsson 1999a). He realised that intrusion-like behaviour is rare compared with non-intrusion-like behaviour, suggesting a ratio of 1:50,000 for audit records. For frames within a correctly configured and secured network, using a defence-in-depth strategy, this ratio could be significantly larger. However, using this ratio a NIS will spend 99.998% of the time processing frames that are either not capable of asserting the security status of the network, or will make the wrong assertion of its security status. In this respect, such a NIS could be considered to be only 0.002% efficient.

The efficiency of a NIS is important when the practical problems of discriminating intrusion from non-intrusion measurements are considered. Signature-matching NIS are able to complete their classification of frames quickly, often in real-time. However when more sophisticated and processor intensive techniques are deployed, such as support vector machines, the inefficiency in the NIS approach can result in substantial increases in the size and cost of the hardware necessary for real-time operation. This will continue to be a problem as the bandwidths increase beyond the 1GB/s networks that are currently widely deployed.

Additional problem areas for passive NIS include (Allen, Christie et al. 1999; Bace and Mell 2001):

- Low-observable intrusion events – A number of intrusion methods are difficult to detect from their network signatures alone. Indeed, the

measurement capability of passive NIS does not guarantee that all network properties can be observed, as described by Monticelli (Monticelli and Wu 1985). A particularly difficult problem is the remote detection of packet sniffers using passive sensing alone. Packet sniffers have only a small effect on the network, due to their passive operation. Very high sensitivity discrimination techniques are required to classify correctly the NIS measurements for detection of packet sniffers;

- The speed and volume of data on the network – As businesses deploy new services onto their networks the difficulties of passive detection become exacerbated. Data intensive applications, such as video streaming or VoIP, will require NIS hardware capable of dealing with the increased data rates. NIS efficiency will worsen as the intrusion-like frames become further diluted within the large number of frames from such applications;
- Separation of data into individual attack streams when multiple intrusions are in progress. This requires discrimination techniques with high selectivity;
- Initialisation of the intrusion system when first deployed in an operational network. When a NIS is first deployed there is likely to be many alerts indicated, causing issues of confidence in the NIS with staff responsible for its maintenance. These alerts are often due to miss-configuration of network devices, applications using non-standard communications techniques and even non-optimal setup of the NIS itself. Although support staff soon learn to recognise the characteristic alerts in

their network that result from these mechanisms, a determined intruder could use these same mechanisms to penetrate further into the network, hiding their activities within perceived normal NIS behaviour;

- Detection of a network interface card (NIC) in promiscuous mode. Many techniques have been identified that are capable of asserting that a NIC is in promiscuous mode by sending special frames (Verwoerd 1999), however detection from passive sensing of network frames alone is problematic;
- Low bandwidth attacks, where a conventional passive NIS could have difficulty in maintaining sufficient state information when the attack occurs slowly, over many days or weeks; and
- Encrypted attacks, in which the intruder encrypts communications, making the extraction of information via passive sensing particularly difficult or almost impossible. Inspection within the payload of an encrypted frame cannot be achieved without access to the encryption keys.

The preceding discussion shows that confidently determining whether a network is under attack can be difficult with a passive NIS just sensing the frames on a network segment. In contrast, active probing deliberately allows the intrusion system to create and transmit special frames over the monitored network, to gain additional information other than that available from passive network sensing. This is discussed further in Appendix D.

4.6. NIS Performance Metrics

In order to develop a deeper understanding of the performance of NISs the

remainder of this thesis will concentrate on performance in terms of sensitivity and selectivity only.

4.6.1. The Problem of Defining Performance

It is not a simple problem to quantify the performance of a NIS. Consider the following example. An intruder launches a network attack on an organisation. The attack consists of $\mathrm{N}_{\mathrm{Attack}}$ frames directed at a single network server. A NIS alerted to $\mathrm{N}_{\mathrm{Alert}}$ frames in the attack, that is multiple signatures were triggered during the attack, where

$$\mathrm{N}_{\mathrm{Alert}} \leq \mathrm{N}_{\mathrm{Attack}}$$

Equation 4-4 Number of Frames in an Attack

What is the probability of detection P_d that should be ascribed to the NIS for the given attack? One answer might be $P_d = 1.0$ as the attack was detected by the NIS. This is a connection-based view of detection, where the detection of an intrusion frame within any of the connection frames classifies the connection as malicious. Alerting against multiple intrusion frames within a single connection is not significant for detection, but could improve attack recognition or identification.

Correlation of the source and destination addresses could allow the other frames in the attack to be identified in subsequent analysis, if full network recording is available (not usually so). Alternatively network recording could have been triggered after the first alert, on the specific source and destination pair of addresses, to allow analysis of frames after the first detection that has occurred. Intrusion Prevention System (IPS) techniques can also be initiated to

prevent further attacks from the same source.

A deeper assessment however, shows this approach to be unsatisfactory. If an alert occurred late in the attack, the post-intrusion damage assessment would be seriously compromised if full network recording was unavailable. Also late detection limits the ability of a NIS to halt the damage caused by an attack. IPS techniques may well stop further intruder activity between the source and destination addresses, but if the attacker has already achieved most of their goals this limitation may not have a significant impact on the attacker. Detections late in a connection are to be expected when malicious payloads are delivered with the intent of causing damage rather than stealing information. This is a limitation of the connection-based view of detection.

An alternative approach to quantifying performance might use the fact that not all of the intrusion frames have been detected, that is:

$$P_d = \frac{N_{Alert}}{N_{Attack}}$$

Equation 4-5 Detection Probability - Frame-Based View

This approach can be considered to be a frame-based performance measure. It is complicated by determining whether or not specific frames were intended as malicious. For example, consider an authorised user logging onto an FTP server, deliberately uploading a malicious payload and then logging off the server. The whole exchange may be over in a small number of frames. The log-on and log-off frames are not malicious as the user is authorised. The upload may use several frames with only one containing malicious data. In this

example it would become difficult to decide which frames should be considered part of the attack. The logon, although by an authorised user was for malicious intent. The frames that did not contain the malicious part of the payload were nevertheless necessary to the attack.

The moment there is a violation of network policy an intrusion event has occurred. The frames involved in the attack prior to the policy violation are of no consequence therefore the simplest definition of P_d has been adopted for the remainder of this thesis.

4.6.2. Sensitivity

Sensitivity represents the ability of a NIS to alert when a specific attack is underway. Its selection as a performance measure was motivated by radio systems, where the noise in a receiver and the environment can limit the reception of a signal. At its simplest level, when the background hiss within the radio exceeds the amplitude of signal being received, it can be difficult to perceive the signal.

The same concept applies in computer networks, where the presence of frames can mask or overwhelm an intrusion system. A simple example of this is alerting on PING. This can be very effective if the network being monitored does not use PING, but ineffective when swamped by the "noise" of normal application-generated or support personal use of PING.

4.6.2.1. Definition of Sensitivity

In detection theory, see for example (Van Trees 2001), the ability of an algorithm to detect the presence of a specific event is cast in terms of two conditional probability density functions that relate to the output from the

algorithm. The first is the probability density function for achieving a given output, x on condition that a valid event, that is an intrusion, is not present, given by $f(x|\overline{intrusion})$. The second probability density function is predicated on the condition that an intrusion is present, given by $f(x|intrusion)$**.** Within many detection systems there is an explicit threshold, t, which can be applied to make the intrusion/no intrusion decision. Under these circumstances and for continuous decision processes in one dimension:

$$P_d = \int_t^{\infty} f(x|intrusion)\, dx$$

$$P_{fa} = \int_t^{\infty} f(x|\overline{intrusion})\, dx$$

Equation 4-6 Definition of P_d and P_{fa}

Where P_d is the probability of declaring that an intrusion is present, on condition that an intrusion is actually present and P_{fa} is the probability of declaring an intrusion to be present when an intrusion is not present.

It is common to represent the probability density functions as Gaussian with standard deviation σ and to represent the conditionality as a difference in the mean, η. Under these circumstances the equations for P_d and P_{fa} become:

$$P_d = \frac{1}{\sigma\sqrt{2\pi}} \int_t^{\infty} e^{\frac{-(x-\eta)^2}{2\sigma^2}} dx$$

$$P_{fa} = \frac{1}{\sigma\sqrt{2\pi}} \int_t^{\infty} e^{\frac{-x^2}{2\sigma^2}}\, dx$$

Equation 4-7 P_d and P_{fa} for Signals in Gaussian Noise

These equations are combined and shown graphically in Figure 4-2. This figure offers a geometric interpretation of detection in terms of the parameters of Equation 4-7. The blue curve shows the probability density function when an intrusion is present whilst the red curve shows it when no intrusion is present.

For fixed η and σ the effect of varying the threshold t can be seen. When t increases both P_d and P_{fa} decrease, with P_d decreasing more rapidly than P_{fa}.

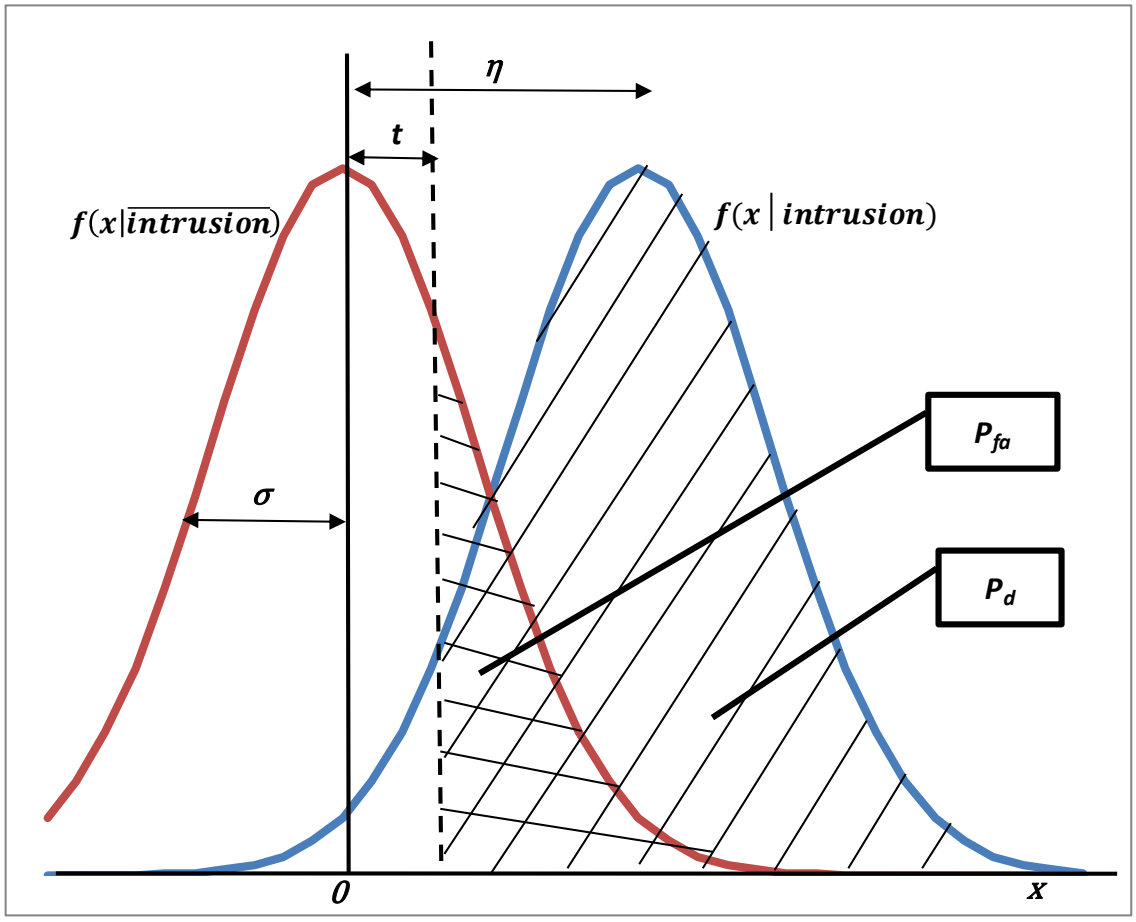


Figure 4-2 Graphical Interpretation of P_d and P_{fa}

There is no value of the threshold, t, that separates out the two conditional probability density functions, unless the conditionality parameter, η, is made arbitrarily large. Indeed the separation of the two conditional probability density functions as a ratio of the width of each density function expresses the degree

of difficulty of detection as a single number. This is the method used in other engineering disciplines to express detection sensitivity, as shown in Equation 4-8 where SNR stands for signal-to-noise ratio measured in decibels (dB):

$$SNR = 20\log(\frac{\eta}{\sigma})$$

Equation 4-8 Definition of Detection Sensitivity

4.6.2.2. Interpretation of Sensitivity

Equation 4-8 defines the detection performance of a NIS in terms of a single parameter, the SNR. Previous measures have required the specification of four parameters namely the false negative, false positive, true negative and true positive rates. With four parameters it is difficult to compare directly differing implementations whilst the use of SNR simplifies comparison; higher values of SNR indicating better detection performance.

In applying Equation 4-8 to network intrusion systems three problems are immediately obvious, that are addressed in the following discussion:

- Conditional probability density functions may not be Gaussian;
- There is no explicit threshold present in a signature-based NIS; and
- η and σ are not normally measured in the evaluation of a NIS.

It is unlikely that the conditional probability density functions will be Gaussian. Although arguments based on the central limit theorem could be applied to imply that they may be Gaussian-like it is better to view the Gaussian assumption as a further parameterisation of the detection model. Alternative parameterisations are possible, for example using a Poisson probability density function. It is also possible to apply this method without knowing the exact

form of the conditional probability density function. The Tchebycheff inequality (Papoulis 1991) can be used to set limits on the detection performance based on η and σ measurements irrespective of the form of the probability density functions. Therefore the effect of a deviation from a Gaussian probability density function could be assessed, in terms of the minimum detection performance implied by the Tchebycheff inequality.

Although there is no specific threshold within signature-based NIS there is an implied threshold setting in the selection of the individual signatures. Consider for example the case when there is just a single signature which is designed to trigger on all possible frames. This is equivalent to setting the threshold, t, at $-\infty$ as both the detection and false alarm probabilities would be 1.0. If the single signature was set to trigger only on improbable frames, such as a frame of only one bit, then this would be equivalent to setting the threshold, t, at $+\infty$ as both the detection and false alarm probabilities would be 0.0. The selection of a given set of real signatures moves the implied setting of the threshold between these extreme values.

More subtly the set of signatures in combination with the network security policy controls the SNR for a given NIS. This can be seen by considering limiting cases of network security policies. If the policy consisted of a single requirement not to use PING within the network then signatures could easily be constructed to trigger only on PING and on every instance. Thus, the P_d would be 1.0 and the P_{fa} would be 0.0, implying a large SNR for the NIS for attacks using PING. If however the network security policy had a single requirement

that passive network sensing, that is packet sniffing, must not be used, it is much more difficult to design signatures to achieve this requirement. Although techniques for the detection of network interface cards in promiscuous mode do exist, achieving this at high SNR is difficult. Missed instances of passive detection are likely (P_d < 1.0) and false alarms are likely to occur (P_{fa} > 0.0). It can be seen therefore that if the security policy clause is difficult to measure with signatures, then that intrusion system will have a low SNR for the corresponding attack method.

Although η and σ are not normally measured in the evaluation of a NIS their ratio can be determined from measurements of P_d and P_{fa}. This can be undertaken using the following algorithm:

1. Set the value of σ to one and determine what threshold setting, t produces the measured P_{fa} ;
2. With η set to zero and σ to one calculate the value of the threshold setting t that produces the measured P_d ; and
3. Determine what value η must be set to align the thresholds from the first two stages. This is the required ratio to be applied in Equation 4-8.

The required absolute value of sensitivity can be interpreted by considering typical network and intrusion statistics. From the data presented in Appendix A for the DARPA 1999 dataset it can be seen that typically, in the network modelled in the DARPA simulation, there were $5x10^4$ connections per day. Given that a reasonable goal might be one false alarm per attack type per day, this would imply a value of P_{fa} of $2x10^{-5}$. A high probability of detection is

desirable (say $P_d = 0.9$) in which case the required sensitivity would need to be 14.6dB. Higher values of sensitivity would be required if the desire was to increase P_d or decrease P_{fa}, as shown in Figure 4-3.

The curves shown in Figure 4-3 were created using a program written in the Mathcad environment rather than using the polynomial relationships for calculating the area under a Gaussian curve (Abramowitz and Stegun 1964). From Figure 4-3 it can be seen that sensitivities greater than 12dB are required for confident detection to occur. Higher values are necessary when the volume of network traffic is greater or when a lower P_{fa} is required.

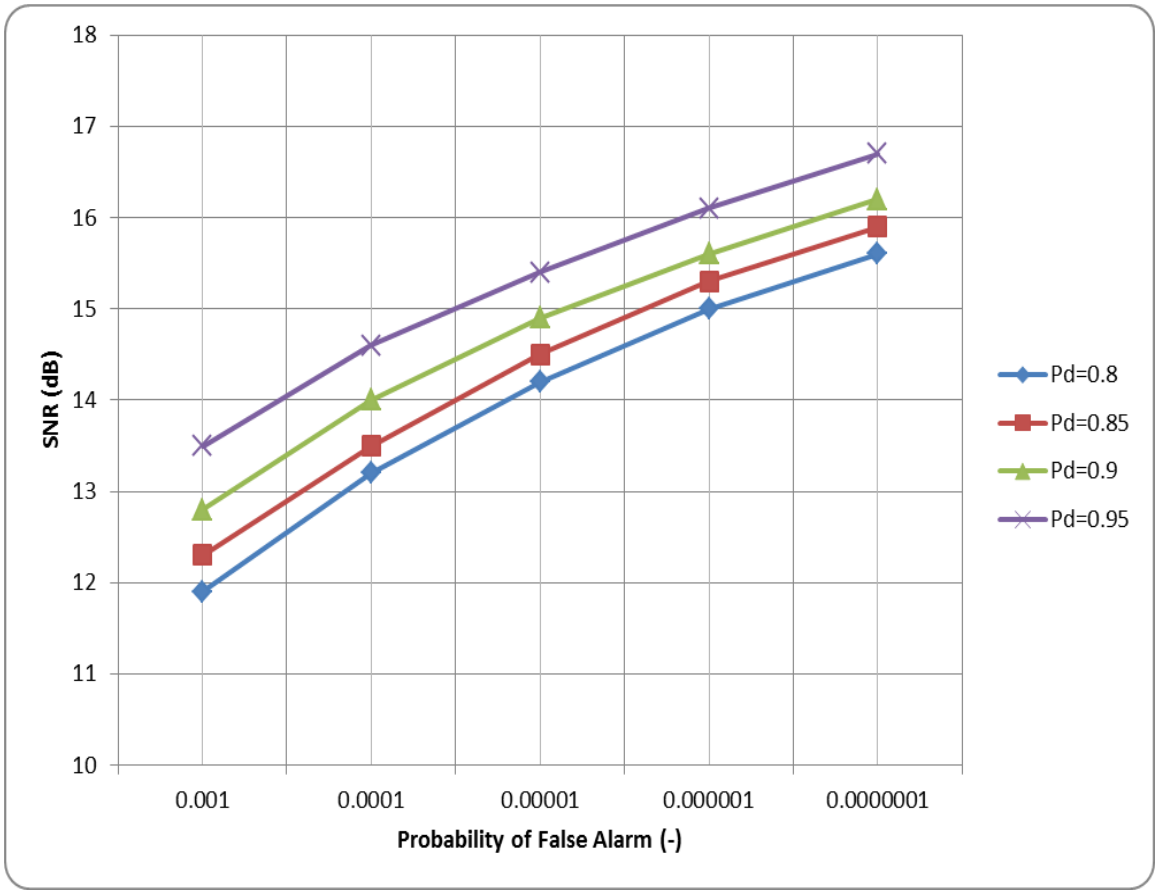


Figure 4-3 Relationship Between P_{fa} and SNR

4.6.3. Selectivity

The use of selectivity as a NIS performance measure was also motivated by its use in radio systems. In such systems, selectivity measures the ability of a receiver to separate adjacent radio channels into distinct signals and not merge them into a single unintelligible one. This is important when the separation of radio channels is small.

Similarly, in NIS selectivity is concerned with the separation of individual attack types. When the number of attacks types is small, equivalent to large channel separation in the radio analogy, distinguishing different attack types may be straight forward. However when the number of attack types is large, this can become more difficult. Selectivity in NIS is concerned with the accurate determination that a given attack type is underway, whilst sensitivity is concerned with determining, in general, that an attack is underway.

4.6.3.1. Definition of Selectivity

Sensitivity is concerned with the detectability of intrusions of interest within the totality of frames present on a network segment. This single measure is useful in quantifying the performance of a NIS in terms of its detection and false alarm statistics. To undertake functions other than detection, such as recognition and identification as discussed in Chapter 3, achieving high sensitivity alone is insufficient. In these cases it is essential to be able to distinguish between different types of attack accurately, so that the correct inferences can be made. This is known as selectivity and it is shown diagrammatically in Figure 4-4 using a pattern recognition paradigm. In this figure clusters for intrusion and non-intrusions are shown plotted on a simplified two-parameter feature space.

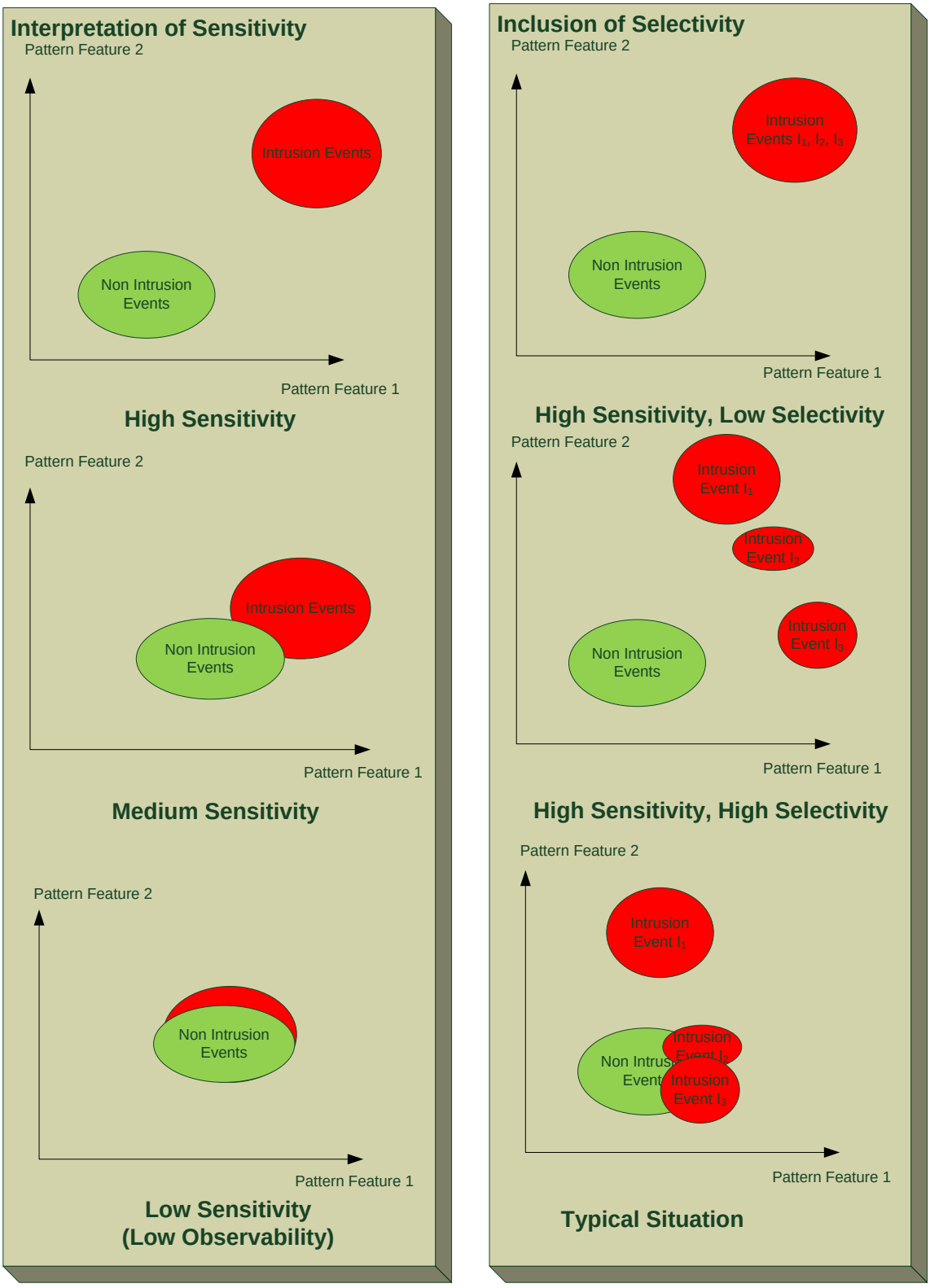


Figure 4-4 Geometric Interpretation of Sensitivity and Selectivity

In the sensitivity column the clusters for different intrusion types are grouped into a single cluster and the difference between low and high sensitivity can be easily seen. In low sensitivity cases the overlap between intrusion and non-intrusion parameters (analogous to S_n and S_i in Equation 4-2) is high, making the correct assertion between intrusion and non-intrusion difficult.

In the column that illustrates selectivity the clusters for different intrusion attacks (events) are shown separately. In conditions of low selectivity there is significant overlap between the individual intrusion clusters making it difficult to confidently assign an event to a specific intrusion attack type. Under conditions of high selectivity it is possible easily to distinguish individual intrusion types. Of particular interest is the "typical situation" graph which shows overlap between some intrusion event types and the non-intrusion events cluster.

In order to develop the measurement of selectivity further it is necessary to consider the set of intrusion event types that need to be discriminated. There are a number of possible sets that could be used:

- Attack types, in which the set members constitute different attack vectors into a network. Clearly distinguishing between set members would identify likely countermeasures;
- Network Policy Requirements, in which each set member is a single or group of requirements. Using this approach compliance issues, as discussed in section 4.2 can be assessed directly; and
- Standardised event types, which would allow direct comparison or benchmarking of different NIS.

Any one of these three types could be used, depending on the application.

The most convenient way of visualising selectivity is as a matrix with the horizontal and vertical axes consisting of the set of intrusion event types to be discriminated. There are a number of different calculations that can be used to quantify the individual entries in the matrix including those based on:

- Confusion matrix values (Provost and Kohavi 1998), in which the number of times the predicted and actual intrusion event types coincide;
- Covariance values (Papoulis 1991), in which the joint probability density function between pairs of intrusion event types is evaluated; and
- Distance values (Duda, Hart et al. 2001), in which each entry quantifies the separation between the two pairs of intrusion events in some parameter space.

In anticipation of the use of the DARPA 1999 dataset for experimental evaluation the distance values approach was selected along with the individual DARPA attack types as the set of intrusion events. The DARPA dataset allows a probability of a given signature being triggered during specific attacks to be determined from their extensive truth data. Therefore a distance measure based in N-dimensional probability space is proposed as follows.

Each intrusion event type (DARPA attack type) has a vector associated with it, the elements of which represent the probability that a given signature will trigger during an attack of that type. Consequently the parameter space is an N-dimensional unity hypercube, where N is the number of signatures used by the NIS, with each intrusion event type represented as a point in this space.

The distance $D(A1, A2)$between two event types A1 and A2 in this space is given the following equation in which $\mathrm{X(i)_{A1}}$ is the probability that the i[th] signature is triggered during an attack of type A1:

$$D(A1, A2) = \sqrt{\sum_{i=1}^{i=N} (X(i)_{A1} - X(i)_{A2})^2}$$

Equation 4-9 Calculation of Selectivity Metric between Two Event Types

This is just the Euclidean distance metric in N-dimensional space. Using Equation 4-9, $D(A1, A2)$ defines selectivity of the two attack types $A1$ and $A2$.

4.6.3.2. Interpretation of Selectivity

The larger the distance $D(A1, A2)$ the easier it is to discriminate the two event types $A1$ and $A2$. As an example consider the following two limiting cases when N is set to 4.

In the first case the signatures that are triggered do not overlap, that is, different signatures are triggered for the two different event types. This can be achieved, for example, by letting $\mathrm{X_{A1}}$={1,1,0,0} and $\mathrm{X_{A2}}$={0,0,1,1}. Under these conditions $D(A1, A2)$ has a value of 2. This represents the perfect discrimination between these two event types.

In the second case the signatures that are triggered overlap perfectly, that is, the same signatures are triggered for both event types, with the same probability level. This can be achieved, for example, by letting $\mathrm{X_{A1}}$={1,1,1,1} and $\mathrm{X_{A2}}$={1,1,1,1}. Under these conditions $D(A1, A2)$ has a value of 0. This represents indistinguishable events.

The theoretical maximum value of D is $\sqrt{\mathrm{N}}$. However, there are only (N-1) pairs

of intrusion event types that can simultaneously take on the maximum value. As there are $N(N-1)/2$ pairs of significance in the selectivity matrix it can be seen that when individual pairs take on the maximum theoretical value it is at the expense of selectivity for other pairs of intrusion event types.

Consider the following situation in which there are 100 signatures and 10 different intrusion event types. If one of the event types has a probability vector whose elements are all unity and all the other vectors have elements of zero, then this event type will have the maximum selectivity of 10 with all the other event types. However, the selectivity between the other event types will be zero and no discrimination between them can occur.

A better approach is to consider the selection of signatures such that the probability vector of each intrusion event type has an equal share of non-overlapping unity values with the vectors of other intrusion event types. That is, each vector has N/M unity values, where M is the number of intrusion event types. In the example above each vector would ideally have 10 elements in their probability vector with a value of unity. Under these conditions the maximum selectivity is $\sqrt{10}$ and this maximum is achievable by all significant pairs in the selectivity matrix.

Thus rather than the theoretical maximum of $\sqrt{N}$ it is better to consider the maximum to be $\sqrt{N/M}$, on the assumption that the discrimination of all pairs of significant intrusion event types are equally important. When this assumption is not valid the number of unity values in the probability matrix for high priority pairs can be increased, to improve selectivity still further.

Finally, in the preceding discussion the emphasis has been on the unity elements of the probability vector for each intrusion event type. This has been done to illustrate the limiting cases and hence the theoretical maximum values of selectivity. In real situations the vectors will contain the full range of probability values, but the goal remains to maximise selectivity and this is best achieved with unity values, that is, signatures should be designed always to trigger when specific intrusion event types occur.

4.7. Metrics for High Level Definitions of Intrusion

The discussion so far has concentrated on the definition of metrics for the "detection" aspect of the taxonomy presented in Chapter 3. It is possible to consider calculating probabilities for recognition (P_R), identification (P_I) and confirmation (P_C).

Sensitivity addresses detection directly whilst selectivity is more concerned with the discrimination between intrusion event types. Sensitivity can be considered as a measure of detection performance, when the number of different attack types is small or even unity, that is there is no attempt to detect specific attack types. However, when sensitivity is applied to a large number of individual attack types, as in the previous discussion, it is more like a performance measure for recognition or identification performance rather than purely detection. Thus sensitivity can apply at higher levels of intrusion functionality depending on the selected attack types.

Clearly, good sensitivity and selectively are a pre-cursor to achieving confident recognition, identification and confirmation. However as algorithms for achieving these goals have not been defined, performance metrics for them will

not be considered further, with the remainder of this thesis concentrating on sensitivity and selectivity without differentiating their application for detection, recognition or identification.

4.8. Conclusions

This chapter has addressed the systems considerations of the operation of a Network Intrusion System (NIS). It started with the development of a mathematical view of NIS operation from the perspective of simple set theory. Currently there is no complete mathematical view of NIS however this chapter has been able to define the properties of an ideal system, in terms of the type of frames present on a network segment. The fundamental problem of signature-based NIS is that some frames can be both intrusion-like and non-intrusion-like simultaneously, making the perfect signature-based NIS unachievable.

Despite this limitation, the properties of an ideal NIS have been discussed after the reasons for deployment were described. From this a functional model of an ideal NIS was developed and current challenges to real systems defined.

Two performance measurements have been identified in terms of a detection model in which intrusion and non-intrusion statistics are parameterised as Gaussian distributions with differing means. These are called as sensitivity and selectivity. The options for alternative parameterisations have been discussed and the potential for lower performance levels described.

Sensitivity defines the performance of an NIS in terms of its ability to alert only when a valid intrusion event is present. To achieve good performance

sensitivities in excess of 12dB are required, depending on the network data rate, required detection probability and acceptable false alarm rate. The use of selectivity applies to detection, recognition and identification performance, depending on the attack types selected for the measurement.

Selectivity defines the ability of a NIS to discriminate between different intrusion event types. It is defined as a distance function between points in a parameter space. The parameter space is spanned by vectors whose elements represent the probability that a specific signature would be triggered, for a given attack type. Theoretical performance properties of selectivity have been derived and the implications for the assignment of signatures to individual clauses of the system security policy have been discussed.

In the following chapter the practical application of sensitivity and selectivity will be described.

CHAPTER 5

EXPERIMENTAL EVALUATION

5. Experimental Evaluation

This chapter describes an experimental evaluation of the performance of a modern signature-based NIS, in terms of the two performance metrics described in the previous chapter. It begins with a discussion of the objectives of the evaluation. Then a description of the experimental configuration is provided, followed by a detailed rational for each of the elements of the configuration. The measured performance of the NIS is then described in detail, first in terms of more conventional metrics and finally in terms of sensitivity and selectivity.

5.1. Objectives of the Experimental Programme

The objectives of the experimental stage of this research were to:

- Provide quantitative performance data on a modern NIS that can be used to highlight areas where improvements are necessary and can be undertaken; and
- Demonstrate the use and value of sensitivity and selectivity in quantifying performance.

It is important to realise that it is not an objective to optimise the performance of the selected NIS. Therefore, although areas of improvement are identified where relevant, the reported performance is not the best that could have been achieved using conventional techniques. No tuning of the NIS has been undertaken beyond identifying the local network segments. Specifically, the standard default set of signatures was deployed in full without any attempt to

remove signatures with high false alarm performance or add ones to detect missed intrusions.

5.2. Overview of the Experimental Configuration

SNORT was selected as the NIS principally due to its maturity. It has been in continual development for over 14 years and during this time has become the core detection engine of many commercial intrusion systems (Sourcefire 2012). There is considerable published material on its performance, optimisation and deployment. Finally, as an Open Source project it is freely available in source code and binary form, allowing modification if required.

A key feature of SNORT is its ability to process previously recorded network frames as well as live frames arriving at a selected network interface card. The use of recorded frames has many distinct advantages over live data for the current research including:

- Repeatability – The measured performance can be confirmed by other researchers;
- Explanation – Unexpected results can be further analysed by examining the recorded frames with tools such as WIRESHARK (Wireshark Foundation 2012); and
- Automation of Results Analysis – With a pre-recorded sequence of network frames it is possible to label specific attacks and therefore automate the collection of performance statistics.

With these advantages in mind the configuration shown in Figure 5-1 was used to gather the results reported in this chapter. As can be seen in this figure,

SNORT uses a set of intrusion signatures and a configuration file to process the recorded network frames. The output from SNORT is written to file and processed offline to extract alerts, which are then compared with a table of truth data containing details on all the real intrusions within the set of recorded network frames. A statistical analysis of the results is then performed from which the performance of the NIS is determined.

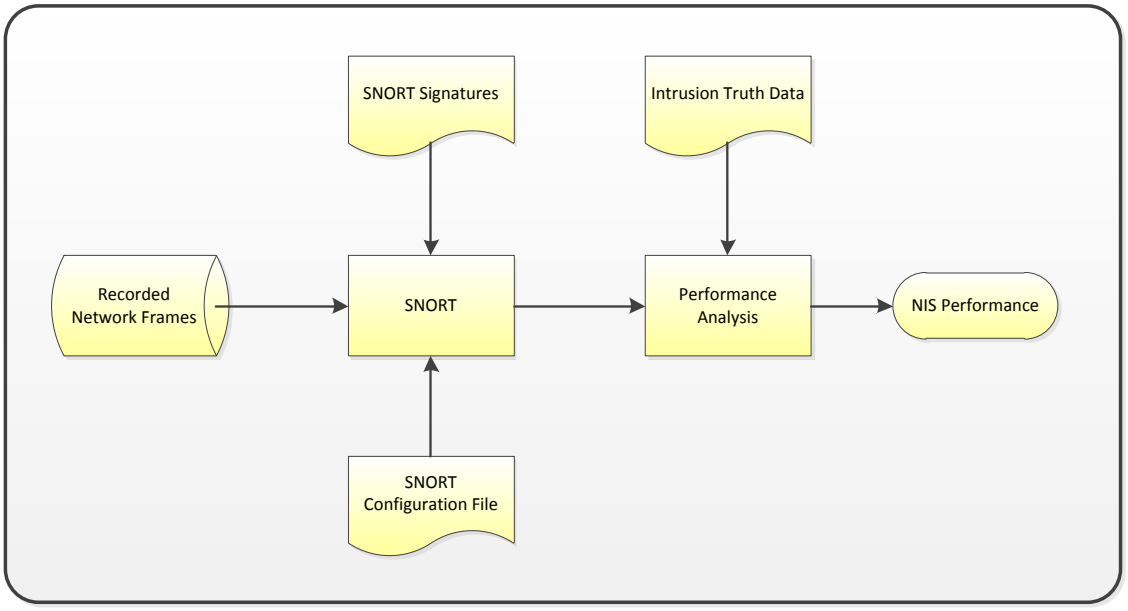


Figure 5-1 Experimental Configuration

5.3. Experimental Setup

5.3.1. Database Selection

There are a number of options available for the database of frames to use with SNORT, as described previously in section 2.6.3. The DARPA 1998-2000 series is of interest as they have been used extensively by other researchers and therefore there is a lot of results which could be contrasted. Also the large TCPDUMP datasets available at CAIDA, RIPE and the WITS project are of

relevance due to the quantity of data and their more modern protocols. However, the DARPA 1999 database was selected, for the following reasons:

- Volume of data – Five weeks of frame data are available covering at least 12 hours each day of a simulated network, totalling over 22GB of data;

- Controlled Environment – As the network traffic is simulated, rather than recorded from a live network, the presence of specific protocols and events is deliberate, rather than accidental. Specifically, two of the five weeks contained no intrusion events, which is particularly useful in determining false positive performance of a NIS. Specific attacks have been introduced at defined instances during weeks 2, 4 and 5; and

- Research Corpus- There is a considerable body of research published using the DARPA 1999 data allowing the comparison of the techniques developed here with that of other researchers.

Although there are compelling reasons to select the DARPA 1999 dataset there are limitations of this data which restrict the conclusions that can be drawn from its use. These were documented in section 2.6.3.2. However, despite these limitations, the lack of modern protocols such as peer-to-peer, VoIP and IM, and the predominance of Unix attacks this database is sufficient to illustrate the use of sensitivity and selectivity as performance metrics.

The DARPA 1999 dataset contains, amongst other things, TCPDUMP files from the inside of the network, that is, inside the simulated network gateway, as well

as separately from outside the gateway. As this increased the volume of data as well as allowing external attacks directed at the network servers to be detected, it was initially intended that both the inside and outside datasets would be used. However during the initial usage of the DARPA data as part of this research it was discovered that the inside and outside data capture machines were not time-synchronised. Figure 5-2 shows the difference between timestamps on identical frames within the inside and outside datasets. These offsets were measured at the beginning of each day of simulation, and do not represent the drift during the day (see Appendix A for a description of the x-axis nomenclature). Appendix B describes the process used to determine the time offset and provides quantitative data on the intra-day clock drift.

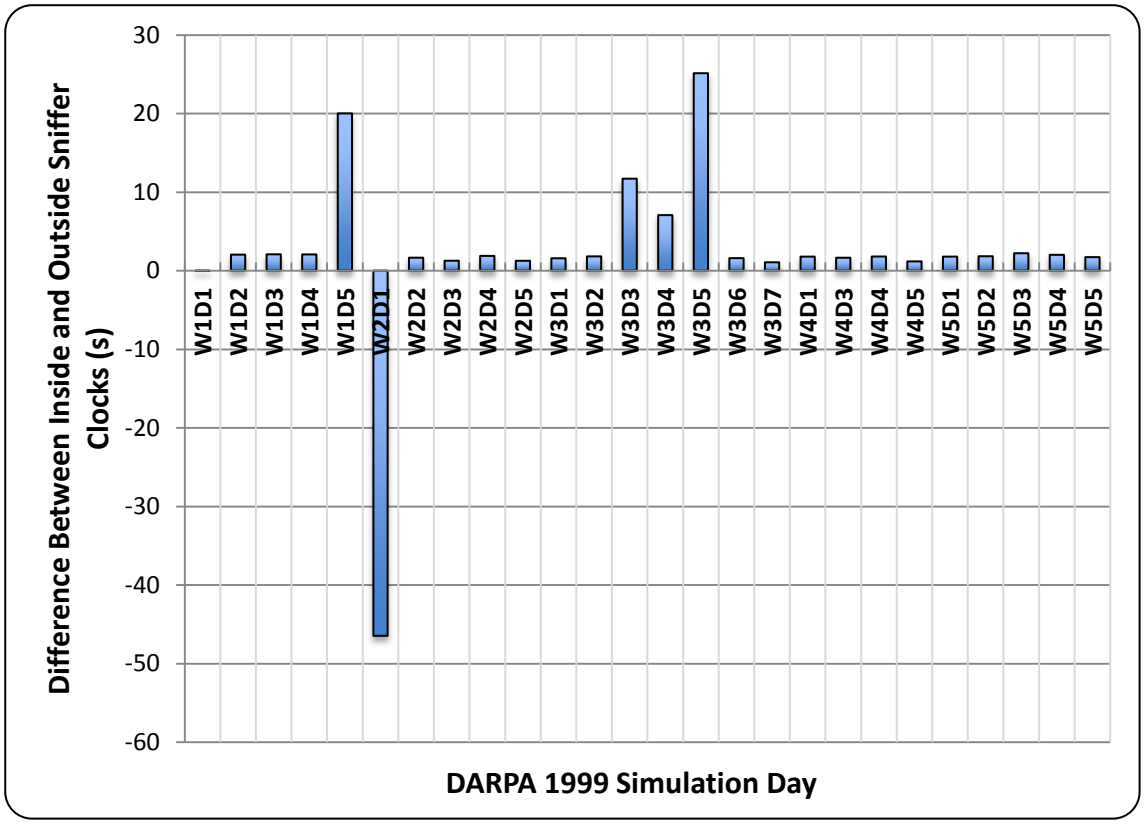


Figure 5-2 Time Synchronisation Error in DARPA 1999

Figure 5-2 shows that time offsets as large as 46 seconds could exist between these two datasets. In order to overcome this problem it would have been necessary to create two sets of truth data, as some of the attacks had durations shorter than the time offsets. Rather than add this additional complexity to the experimental work it was decided to limit the assessment of performance to that achieved against the DARPA TCPDUMP data taken inside the network only.

5.3.2. Intrusion Truth Data

The DARPA 1999 dataset includes a set of intrusion truth data from which correct intrusion alerting could be determined. There are two types of truth data, one each for the training and testing datasets.

5.3.2.1. Training Dataset Truth Data

During Week 2 of the DARPA 1999 simulation 43 deliberate intrusion events were included. It was intended that this data, along with the intrusion event free weeks 1 and 3, would be used as training data for intrusion systems that required it.

The truth data for this week consists of the date and time of the start of the attack, the DNS name of the target of the attack and the type of attack, including a description of the way the attack was implemented. The IPv4 address of the source of the attack and the attack duration were not recorded.

The lack of information on the attack duration is a serious limitation of this data. The matching of intrusion alerts with attacks requires accurate time and duration of these events, however no additional data other than that published on the Lincoln Labs website was available (Lippmann 2008). Some researchers

have assumed that the duration of the attacks was one second (Mahoney 2012).

During this research the original DARPA truth data was supplemented with measurements taken using WIRESHARK (see Appendix C). Of the 43 attacks during week 2 evidence for six of them could not be located from inspection of the recorded network frames and were therefore excluded from this research. A histogram of the attack durations for those that could be detected is shown in Figure 5-3. As can be seen the one second assumption is inaccurate with 13 attacks having durations of over 200 seconds.

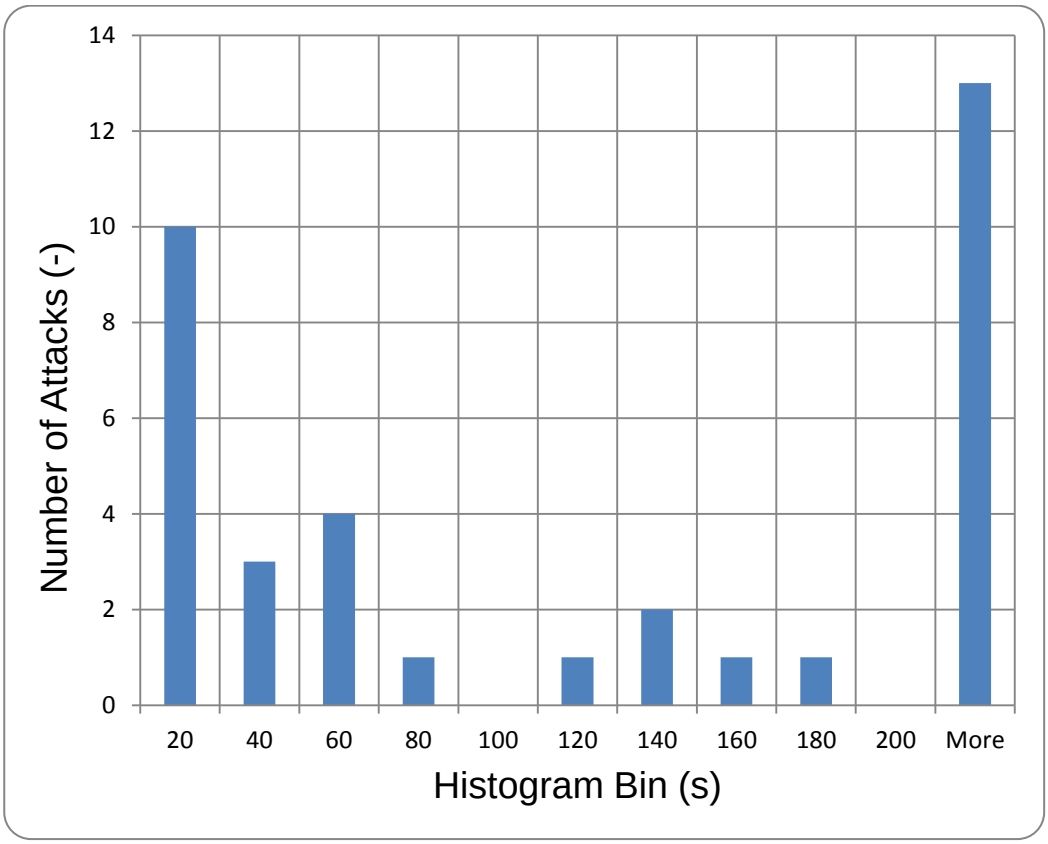


Figure 5-3 Histogram of Attack Durations for Week 2 of DARPA 1999

5.3.2.2. Testing Dataset Truth Data

The testing truth data was more comprehensive than the training dataset truth. Table 5.1 shows the parameters recorded for each attack, along with a description of their interpretation. As the DARPA data was collated for evaluation of both host-based and network intrusion systems, an additional column is shown to indicate if the truth data parameter applies to NIS evaluation. Despite this comprehensive list of data, the IPv4 address of the source of the attack was not recorded.

Parameter	Description	For NIS
IDum	Identity number for a given attack instance.	Yes
Date	Date of the attack.	Yes
StartTime	Start of the attack.	Yes
Duration	Duration of the attack.	Yes
Destination	IPv4 address of the victim of the attack.	Yes
Attackname	The common name of the attack type.	Yes
Insider	Indicating if the attack originated from a host inside or outside the DARPA network.	Yes
Man	Indicating if the attack was manually initiated at a console or automated through a script.	Yes
Console	Indicating if the attack was carried out on the console of the target machine or undertaken remotely.	Yes
Success	Indicating if the attack was successful or not.	Yes
aDump	Indicating if there was a host dump file of the attack.	No
oDump	Indicating that there was evidence of the attack in the outside TCPDUMP files.	Yes
iDump	Indicating that there was evidence of the attack in the inside TCPDUMP files.	Yes
BSM	Indicating if there is evidence of the attack in the solaris BSM log.	No
Syslogs	Indicating if there is evidence of the attack in the system logs.	No

Parameter	Description	For NIS
FSListing	Indicating if there is evidence of the attack in the file system data.	No
Stealthy	Indicating if the attack is considered stealthy or not.	Yes
New	If the attack was new to the DARPA 1999 evaluation, that is, it was not in the DARPA 1998 simulation.	Yes
Category	Indicating the category of the attack.	Yes
OS	Indicating the operating system of the attack victim.	Yes

Table 5-1 Testing Truth Data for DARPA 1999

5.3.3. SNORT Configuration and Signature Files

The actions of SNORT are controlled via a configuration file. Appendix E shows the configuration file used for all the results published in this thesis. In this Appendix, the comments have been deleted to reduce the size of the final text.

The configuration file was modified from the standard one supplied with SNORT, in the following ways:

- The home network was set to the internal network and server subnet of the DARPA dataset (inside);
- Directory paths were selected to allow SNORT to import signatures and additional standard data files from specified locations;
- The output format for alerts was selected as comma separated variable (CSV) with full logging of intrusion data; and
- All standard intrusion signatures were enabled.

It is usual for SNORT to undergo an optimisation process when first deployed within a network, by selecting and enabling a subset of the available intrusion signatures. The aim of this optimisation is to eliminate non-intrusion alerts that can arise from normal network activity, rather than from intruder activity. This

was not undertaken for the experimental work reported here so that the raw performance of SNORT could be determined. Also the elimination of common alert types on a network is an unsatisfactory approach to poor false alarm performance as it opens an attack vector for an intruder.

Table 5.2 shows the version types for SNORT and the supporting software that was used during this experimental programme.

Software	Version
SNORT	2.8.6 Dated 26th April 2010
SNORT Rules	2860, Dated 13th May 2010
LIBPCAP	1.1.1
PCRE	8.02
Linux OS	Fedora 13

Table 5-2 Software Versions Used in the Experimental Work

SNORT, LIBPCAP and PCRE were downloaded as source from their primary websites and compiled to run on the Linux workstation on which all the experimental work was undertaken.

5.3.4. Truth Data and Performance Analysis

Shell scripts were written to automate the running of SNORT against each of the inside TCPDUMP files, extracting the results from each and collating them into single files for the whole of the five weeks of simulated data. Three consolidated sets of results were produced covering:

- A count of the number of each signature type that was triggered;

- A summary of the number of frames and sessions processed, along with the number of signatures triggered for each of the DARPA data files (simulation days); and
- A single file combining all the data associated with every signature triggered, including time, source and destination IPv4 addresses and ports used, in CSV format.

The next stage was to match each alert produced by SNORT against an event in the truth table, labelling the detection as a true or false positive, depending on whether or not a match occurred. At first the solution to this problem seemed straight forward. All that was necessary was to create a program to check the dates and times of alerts, as well as target IP address, network protocol and target port against the published truth data.

A PERL script was created to achieve this goal but the number of matched detections was unexpectedly low. The truth table events did not match many with the signatures that had been triggered. The matching criteria were reduced to date, time and target IP address, but the number of alerts matching events in the truth data was still low. The PERL script was further modified to allow a fuzzy match for the time of the event, as all other parameters in the matching algorithm were precise. The low number of matches persisted.

In an attempt to confirm that the poor performance was real the individual TCPDUMP files were examined frame by frame using WIRESHARK, for evidence of individual attacks. This was a major task as can be seen from the summary data presented in Appendix A. The inside dataset consists of over 50 million

frames across 26 files, with many of the files too large to be loaded into WIRESHARK.

In order to address these problems, WIRESHARK filters were developed to extract relevant frames from the TCPDUMP files, which were relevant to specific attacks. The use of WIRESHARK in this way is described in Appendix C, along with the definition of the filters necessary to highlight one specific attack, namely NTinfoscan. This approach reduced the number of frames viewed within the WIRESHARK frame window, simplifying the process of visually identifying the frames within an intrusion event. One outcome from this process was the revelation of the time synchronisation problem shown in Figure 5-2.

This process identified a number of differences between the official truth data and that revealed by the examination of the individual attack frames. The most significant difference concerned the start time and duration of each attack. Many of these were incorrectly recorded in the original truth data, with errors exceeding the duration of many attacks. A new truth table was produced using the WIRESHARK analysis, combining the attacks in the training and testing datasets. This table differed from the original truth data in a number of ways:

- All events are now recorded in UTC. The original truth data recorded events in EST and was further complicated by two changes to daylight saving time during the five weeks of simulated network activity. The UK changed to BST on 28th March affecting events in weeks 4 and 5, whilst the US changed on 4th April, affecting only week 5. As SNORT processes individual frames it converts the TCPDUMP timestamp stored with each

frame to the local time standard by default. SNORT detection events are then recorded in the local time standard which can include the effects of the change to daylight saving time on the computer on which it is running. As time was used to correlate alerts with the truth data it was necessary to take steps to avoid the potential for timing errors due to the one or two hour time slips. SNORT was forced to record all events in UTC via a command line option and the truth data was presented in the same time system; and

- A complete set of data was not always present in the original truth data, as described in section 5.3.2 for Week 2. The enhanced truth dataset was supplemented with missing data, such as the source IPv4 address of the attacker and attack duration.

Although the move to UTC was compelling and simplified the matching process it did introduce an additional complication that each day in the original DARPA simulation is now spread across two days. Therefore date and time of intrusion events were now required to match with the appropriate truth data entries.

Figure 5-4 shows an analysis of this final truth data showing the distribution of attack durations. It can be seen that although the majority of attacks are over in less than one minute, nearly 10% of attacks last more than 20 minutes.

With the truth data and the SNORT alerts both recorded in UTC, the alerts can be split into those that match attacks, that is true positives, and those that do not. Although scripts had been created originally to automate the matching process, as described above, the final split was made manually. This unusual

step was taken as the split was to be undertaken once and the time taken was shorter than the time to finalise and test the scripts using the new truth data. As correct classification of alerts is key to this research, a manual assignment removed any uncertainty in this process. In view of the difficulties that had been experienced with automating scripts this seemed the most prudent approach.

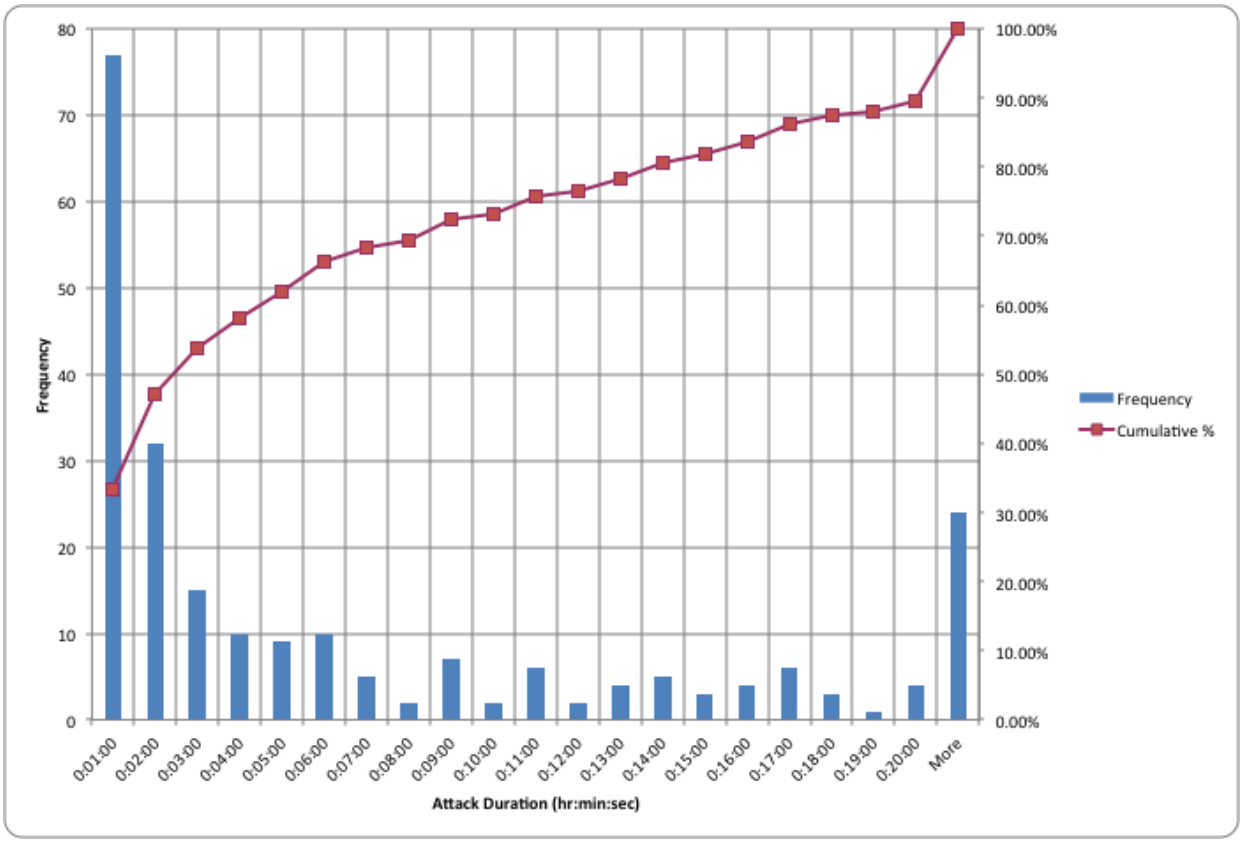


Figure 5-4 The Duration of Attacks in DARPA 1999

5.4. Results

5.4.1. Alert Statistics

Using the configuration file shown in Appendix E SNORT was run against the DARPA 1999 TCPDUMP files that were collected from inside the simulated network. In all five weeks, 67,384 alerts were produced as shown in Table 5.3 below.

Week	No of Alerts	Intrusions Present
1	9404	No
2	25138	Yes
3	12248	No
4	8518	Yes
5	12076	Yes

Table 5-3 SNORT Intrusions Detected

The results from Weeks 1 and 3 can be used to estimate the false positive rate for SNORT processing the DARPA 1999 dataset, as shown in Table 5.4 below, using frame statistics derived from Appendix A and those shown in Table 5.5. False alarms per frame, per connection and per second are shown as the original DARPA 1999 data is simulated and therefore these measures can be used to gauge the size of the simulation.

Week	False Alarms Per Frame (-)	False Alarms Per Connection (-)	False Alarms Per Second (-)
1	1.19×10^{-3}	3.52×10^{-2}	2.37×10^{-2}
3	9.56×10^{-4}	3.46×10^{-2}	2.32×10^{-2}
Average	1.05×10^{-3}	3.48×10^{-2}	2.35×10^{-2}

Table 5-4 SNORT False Positive Performance

Table 5.4 highlights the poor performance of SNORT when it is not optimised for a given network. On average one in every thousand frames is incorrectly declared as an intrusion alert. Given the number of frames per second this means a false positive is declared every 42 seconds, or 85 false positives per

hour. Performance this poor means that manual assessment of the alerts would be impractical and automated post alert assessment would be required.

Week	No of Frames (-)	No of Connections (-)	No of Seconds (-)
1	7,887,003	267,141	395,991
3	12,814,738	354,272	527,024
Total	20,701,741	621,413	923,015

Table 5-5 DARPA 1999 Networking Statistics

5.4.2. False Alarm Assessment

By matching the alerts with intrusions in the truth data they can be classified as a false or true positive. Table 5.6 shows the total count of each of the SNORT signatures that did not match with an intrusion event, across all five weeks of the DARPA simulation. This table shows some interesting results. First, as there were a total of 67,384 alerts produced, over 73% did not match any intrusion event in the DARPA attack simulations. The false positives are dominated by ICMP signature events. By removing these signatures from SNORT the number of false alarms would reduce from 49,458 to 21,380. However, the IP and port sweeps which use ICMP pings would not be detectable.

Secondly, only 36 different signatures were triggered as false positives. During the analysis SNORT was using 3,321 different signatures to search for intrusions therefore such a small number of different signatures producing false positives is surprising. Across all 67,384 alerts, including the true positives, only 52 different signatures were triggered.

At first sight the small number of different signatures might be thought to be

due to the age difference between the signatures and the TCPDUMP data. In the 11 years between the DARPA simulation and the signatures used by SNORT there has been considerable expansion of the number and type of protocols used in modern networks. However, other researchers have reported similar results processing real data captured from a network in over 40 days in 2008 (Tjhai, Papadaki et al. 2008).

SNORT Detection Signature	Total False Alerts
ICMP Destination Unreachable Port Unreachable	17656
(ftp_telnet) FTP command parameters were malformed	16328
ICMP Echo Reply	5088
ICMP PING	5036
CHAT IRC message	1655
TELNET login incorrect	740
ATTACK-RESPONSES directory listing	617
(spp_ssh) Protocol mismatch	469
ICMP PING BSDtype	298
ICMP PING *NIX	298
(ftp_telnet) Invalid FTP Command	281
SHELLCODE x86 NOOP	201
CHAT IRC nick change	193
CHAT IRC channel join	182
ICMP Destination Unreachable Host Unreachable	85
SHELLCODE x86 inc ebx NOOP	81
SHELLCODE x86 inc ecx NOOP	55

SNORT Detection Signature	Total False Alerts
ICMP Time-To-Live Exceeded in Transit	51
(ftp_telnet) FTP traffic encrypted	39
FTP Bad login	27
WEB-CLIENT Portable Executable binary file transfer	13
(http_inspect) NON-RFC DEFINED CHAR	12
WEB-CLIENT Microsoft emf metafile access	11
(ftp_telnet) Evasive (incomplete) TELNET CMD on FTP Command Channel	9
FTP PORT bounce attempt	8
SHELLCODE x86 setuid 0	5
FTP passwd retrieval attempt	4
(ftp_telnet) Telnet Subnegotiation Begin Command without Subnegotiation	3
NETBIOS SMB C$ unicode share access	2
NETBIOS SMB D$ unicode share access	2
X11 xopen	2
(ftp_telnet) Telnet traffic encrypted	2
NETBIOS SMB ADMIN$ unicode share access	2
SQL ping attempt	1
SQL version overflow attempt	1
WEB-MISC cat%20 access	1
Grand Total	49,458

Table 5-6 False Positive Alert Types

Table 5.7 shows the complete set of detection signatures, along with the number of times the signature created true and false positives. In this context a true positive is when a signature was triggered during a valid attack, and with the same source and destination IP addresses as in the simulated attack. A false positive was declared when an alert was triggered outside the duration of a simulated attack, or during an attack with one or other of the source and destination IP addresses not corresponding with that used in the simulated attack.

Detection Signature	No. False Alerts	No. True Alerts	P(correct alert)	P(false alert)
WEB-MISC cat%20 access	1	5	0.833	0.167
X11 xopen	2	8	0.800	0.200
ATTACK-RESPONSES directory listing	617	133	0.177	0.823
SHELLCODE x86 inc ebx NOOP	81	0	0.000	1.000
SHELLCODE x86 inc ecx NOOP	55	27	0.329	0.671
SNMP request tcp	0	87	1.000	0.000
SNMP trap udp	0	3	1.000	0.000
SNMP trap tcp	0	87	1.000	0.000
SNMP AgentX/tcp request	0	86	1.000	0.000
CHAT IRC message	1655	0	0.000	1.000
WEB-CLIENT Portable Executable binary file	13	0	0.000	1.000
POLICY potentially executable file upload	0	13	1.000	0.000
CHAT IRC channel join	182	0	0.000	1.000

Detection Signature	No. False Alerts	No. True Alerts	P(correct alert)	P(false alert)
IMAP login buffer overflow attempt	0	2	1.000	0.000
SQL ping attempt	1	0	0.000	1.000
SQL version overflow attempt	1	0	0.000	1.000
WEB-CLIENT Microsoft emf metafile access	11	0	0.000	1.000
NETBIOS SMB D$ unicode share access	2	3	0.600	0.400
NETBIOS SMB C$ unicode share access	2	3	0.600	0.400
NETBIOS SMB ADMIN$ unicode share access	2	3	0.600	0.400
FINGER / execution attempt	0	24	1.000	0.000
FINGER root query	0	4	1.000	0.000
FINGER redirection attempt	0	4	1.000	0.000
FINGER 0 query	0	4	1.000	0.000
FTP .rhosts	0	4	1.000	0.000
FTP PORT bounce attempt	8	6	0.429	0.571
FTP passwd retrieval attempt	4	0	0.000	1.000
FTP satan scan	0	3	1.000	0.000
ICMP PING *NIX	298	2	0.007	0.993
ICMP PING BSDtype	298	2	0.007	0.993
ICMP PING	5036	7076	0.584	0.416
ICMP Destination Unreachable Host Unreachable	85	0	0.000	1.000
ICMP Destination Unreachable Port Unreachable	17656	2816	0.138	0.862

Detection Signature	No. False Alerts	No. True Alerts	P(correct alert)	P(false alert)
ICMP Echo Reply	5088	2	0.000	1.000
ICMP Time-To-Live Exceeded in Transit	51	0	0.000	1.000
ICMP PING NMAP	0	7000	1.000	0.000
FTP Bad login	27	80	0.748	0.252
CHAT IRC nick change	193	0	0.000	1.000
RPC portmap listing TCP 111	0	199	1.000	0.000
RSERVICES rlogin login failure	0	1	1.000	0.000
SHELLCODE x86 NOOP	201	23	0.103	0.897
SHELLCODE x86 setuid 0	5	0	0.000	1.000
SHELLCODE Linux shellcode	0	2	1.000	0.000
TELNET login incorrect [**]	740	156	0.174	0.826
(http_inspect) NON-RFC DEFINED CHAR	12	0	0.000	1.000
(ftp_telnet) Invalid FTP Command	281	4	0.014	0.986
(ftp_telnet) FTP command parameters were malformed	16328	54	0.003	0.997
(ftp_telnet) FTP traffic encrypted	39	0	0.000	1.000
(ftp_telnet) Evasive (incomplete) TELNET CMD	9	0	0.000	1.000
(ftp_telnet) Telnet traffic encrypted	2	0	0.000	1.000
(ftp_telnet) Telnet Subnegotiation Begin Comm	3	0	0.000	1.000
(spp_ssh) Protocol mismatch	469	0	0.000	1.000

Table 5-7 Analysis of Signatures Triggered by DARPA 1999

It can clearly be seen that a number of signatures, such as "SHELLCODE x86

inc ebx NOOP" only triggered during non-intrusion events. However other signatures, such as "ATTACK-RESPONSES directory listing" produce both real and false positives. Some signatures, such as all those related to SNMP, only produced true positives.

Also shown in Table 5.7 are estimates of the a priori probabilities of correct and false alerts being generated by each SNORT signature. These were derived from the simple relations

$$P(correct\ alert) = \frac{No.True\ Alerts}{(No.True\ Alerts + No.\ of\ False\ Alerts)}$$

$$P(false\ alert) = \frac{No.False\ Alerts}{(No.True\ Alerts + No.False\ Alerts)}$$

Equation 5-1 Estimation of the A Priori Statistics for Each SNORT Signature

5.4.3. Detectability of Attack Types

Table 5-7 shows the detection performance from the perspective of different SNORT signatures. Of more direct interest for the current research is detection performance in terms of the different attack types simulated in the DARPA 1999 dataset. This is shown in Table 5-8.

Type	Attack Type	No. of Attacks	No. Detected	P_d
Denial of service	Apache2	3	0	0.000
	Arppoison	4	0	0.000
	Back	6	0	0.000
	Crashiis	10	0	0.000
	Dosnuke	4	0	0.000

Type	Attack Type	No. of Attacks	No. Detected	P_d
	Land	4	0	0.000
	Mailbomb	6	0	0.000
	SYN Flood	6	4	0.667
	Ping of Death	6	4	0.667
	ProcessTable	2	0	0.000
	Selfping	3	0	0.000
	Smurf	5	0	0.000
	Sshprocesstable	1	0	0.000
	Syslogd	4	0	0.000
	Tcpreset	3	0	0.000
	Teardrop	3	0	0.000
	Udpstorm	2	0	0.000
	Warezclient	4	4	1.000
User to Root (U2R)	Anypw	1	0	0.000
	Casesen	3	3	1.000
	Eject	5	2	0.400
	Ffbconfig	3	0	0.000
	Fdformat	2	1	1.000
	Loadmodule	5	1	0.200
	Ntfsdos	3	0	0.000
	Perl	8	1	0.125
	Ps	2	0	0.000
	Sechole	2	2	1.000
	Xterm	3	3	1.000

Type	Attack Type	No. of Attacks	No. Detected	P_d
	Yaga	4	4	1.000
Remote to Local (R2L)	Dictionary	1	1	1.000
	Framespoofer	1	1	1.000
	FTPwrite	4	4	1.000
	GuessFTP	2	2	1.000
	GuessPOP	1	0	0.000
	GuessTELNET	4	3	0.750
	Guest	3	3	1.000
	HTTPTunnel	5	0	0.000
	IMAP	2	2	1.000
	Named	3	3	1.000
	NCFTP	5	5	1.000
	Netbus	4	3	0.750
	Netcat	4	4	1.000
	Phf	5	5	1.000
	Ppmacro	3	3	1.000
	Sendmail	2	2	1.000
	SNMPget	4	0	0.000
	SQLAttack	2	0	0.000
	SSHTrojan	3	0	0.000
	Xlock	3	3	1.000
	Xsnoop	3	3	1.000
Probes	Insidesniffer	2	0	0.000
	IPSweep	10	9	0.900

Type	Attack Type	No. of Attacks	No. Detected	P_d
	LSDomain	2	0	0.000
	Mscan	1	1	1.000
	NTinfoscan	4	4	1.000
	Portscan	18	5	0.278
	Questo	4	0	0.000
	Resetscan	1	0	0.000
	Satan	4	4	1.000
Data	Secret	8	2	0.250

Table 5-8 Detectability of Different Attack Types

The different attack types are defined in Appendix A. The "No. of Attacks" column shows the number of different attacks present in the enhanced truth data for the DARPA simulation, that is, the number of unique attack labels that pertain to the specific attack type. The "No. Detected" column shows the number of distinct attacks for which there was a SNORT alert issued within the duration of the attack, and with the same source and destination IP addresses. There was no correlation of the signature that was triggered with the attack mechanism in use and therefore the potential for accidental detection is present due to signatures being triggered that were not related to the attack. Indeed, some alerts occurred due to the presence of ICMP PINGs rather than attack specific signatures.

Table 5-8 shows some interesting results. There are many attack types that are not detected at all whilst others are easily detected. Only two attack types (Perl

and Portscan) are marginally detected, that is with P_d near to zero. Examination of the Perl result in detail suggests that the single detection was due to a noisy SNORT signature ("(ftp_telnet) FTP command parameters were malformed" of which there were 16,382 instances during the simulation). Only 36 different SNORT signatures account for all of the detections in Table 5-8.

5.4.4. Sensitivity Measurements

The measurement of sensitivity can be illustrated by examining two specific attacks, for example FTPWrite and Xterm. Table 5-8 shows that both are easily detectable with a P_d of 1.0. However, the sensitivity approach yields a different result, showing that SNORT is not sensitive enough to detect Xterm with any confidence.

Consider Table 5-9 in which the individual signatures that are triggered for each of these attacks are shown. As can be seen there are only two different signature types that are responsible for detecting these two attacks. In each attack simulation each of the attack types is consistently detected by the same signatures, giving some confidence that both attack types are being detected properly. However there were 16,328 instances where the "(ftp telnet) FTP command parameters were malformed" triggered when there was no attack underway. Therefore the presence of this alert conveys less useful information than the "FTP .rhosts" signature which only occurred during real attacks.

Attack Type	Attack Label	Triggered SNORT Signatures
FTPWrite	31.000000	(ftp_telnet) FTP command parameters were malformed
		FTP .rhosts

	43.000000	(ftp_telnet) FTP command parameters were malformed
		FTP .rhosts
	41.135830	(ftp_telnet) FTP command parameters were malformed
		FTP .rhosts
	52.101901	(ftp_telnet) FTP command parameters were malformed
		FTP .rhosts
Xterm	52.100738	(ftp_telnet) FTP command parameters were malformed
	55.091529	(ftp_telnet) FTP command parameters were malformed
	55.174733	(ftp_telnet) FTP command parameters were malformed

Table 5-9 SNORT Signatures for FTPWrite and Xterm

This qualitative argument can be quantified by the following. Consider an attack type consisting of M individual attacks within the DARPA dataset. For each attack let there be N signatures triggered, including multiple triggers of the same signature. Let $\mathrm{P_{fa}}(\mathrm{sig}, \mathrm{atk})$ be the probability of an individual signature, sig, in a given attack, atk taken from Table 5-7 being triggered. The probability of false alarm of an attack type $\widehat{\mathrm{P}}_{\mathrm{fa}}$ and probability of detection, $\widehat{\mathrm{P}}_{\mathrm{d}}$ for the attack type is given by:

$$\hat{P}_{fa} = \prod_{atk=1}^{M} \prod_{sig=1}^{N} P_{fa}\,(sig, atk)$$

$$\hat{P}_d = 1 - \hat{P}_{fa}$$

Equation 5-2 Estimation of $\widehat{\boldsymbol{P}}_{\boldsymbol{d}}$ and $\widehat{\boldsymbol{P}}_{\boldsymbol{fa}}$ for an Attack type

In deriving Equation 5-2 it was assumed that an alert is declared for a given attack type when any one of the signatures is triggered. In this respect it can

be considered as an OR logic rather than an AND logic in which all the signature types must trigger to declare an attack type to be present.

For the FTPWrite and Xterm attack types, this results in Table 5-10. The values of $\hat{P}_d$ and $\hat{P}_{fa}$ from this table can be placed into the algorithm discussed in section 4.6.2 to yield a SNR for the FTPWrite attack type of 18.6dB and for the Xterm attack type of 0dB. Thus this SNORT implementation is more sensitive to detecting FTPWrite attack types than detecting Xterm attacks, in contradiction of the results implied by Table 5-8.

Attack Type	M	N	Pfa(1,atk)	Pfa(2,atk)	$\hat{P}_d$	$\hat{P}_{fa}$
FTPWrite	4	2	0.997	0	1	0
Xterm	3	1	0.997	-	0.008	0.992

Table 5-10 $\hat{P}_d$ and $\hat{P}_{fa}$ for the Attack Type FTPWrite and Xterm

To automate this process a Microsoft Excel spreadsheet was created to take the results from the analysis of SNORT against each attack type and calculate the corresponding $\hat{P}_d$ and $\hat{P}_{fa}$ values. This spreadsheet did not use the individual signature P_d and P_{fa} values from Table 5-7. Instead modified values were used in which the "number of true alerts" was reduced only to those that applied to a given attack type. Consequently the $\hat{P}_{fa}$ values were worse than used in the method outlined above for FTPWrite and Xterm with a corresponding reduction in sensitivity.

The calculation of SNR from $\hat{P}_d$ and $\hat{P}_{fa}$ values was undertaken by creating a Mathcad program. This was necessary to calculate the appropriate areas under

Gaussian functions. In order to do this, values of 1 and 0 were replaced with 0.99999 and $1x10^{-5}$ respectively, due to the accuracy of numerical integration.

When this method is applied to all the attack types in the DARPA dataset the result is shown in Figure 5-5. This figure shows that there are still many attack types that are not detectable and some, such are Dictionary, that are easily detectable. Only nine of the attack types achieve the 12dB level described in the previous chapter.

5.4.5. Selectivity Measurements

The measurement of selectivity is concerned with discriminating between different attack types. The raw data for each signature triggered in the SNORT simulation is shown in Table 5-11 as a function of the different attack types. In this table blank entries indicate that the specific signature did not trigger for the corresponding attack type. Signatures that did not trigger during any of the attacks are not shown, as are attack types in which no signatures were triggered. This was done to reduce the size of the table and therefore present only relevant combinations of signature and attack type.

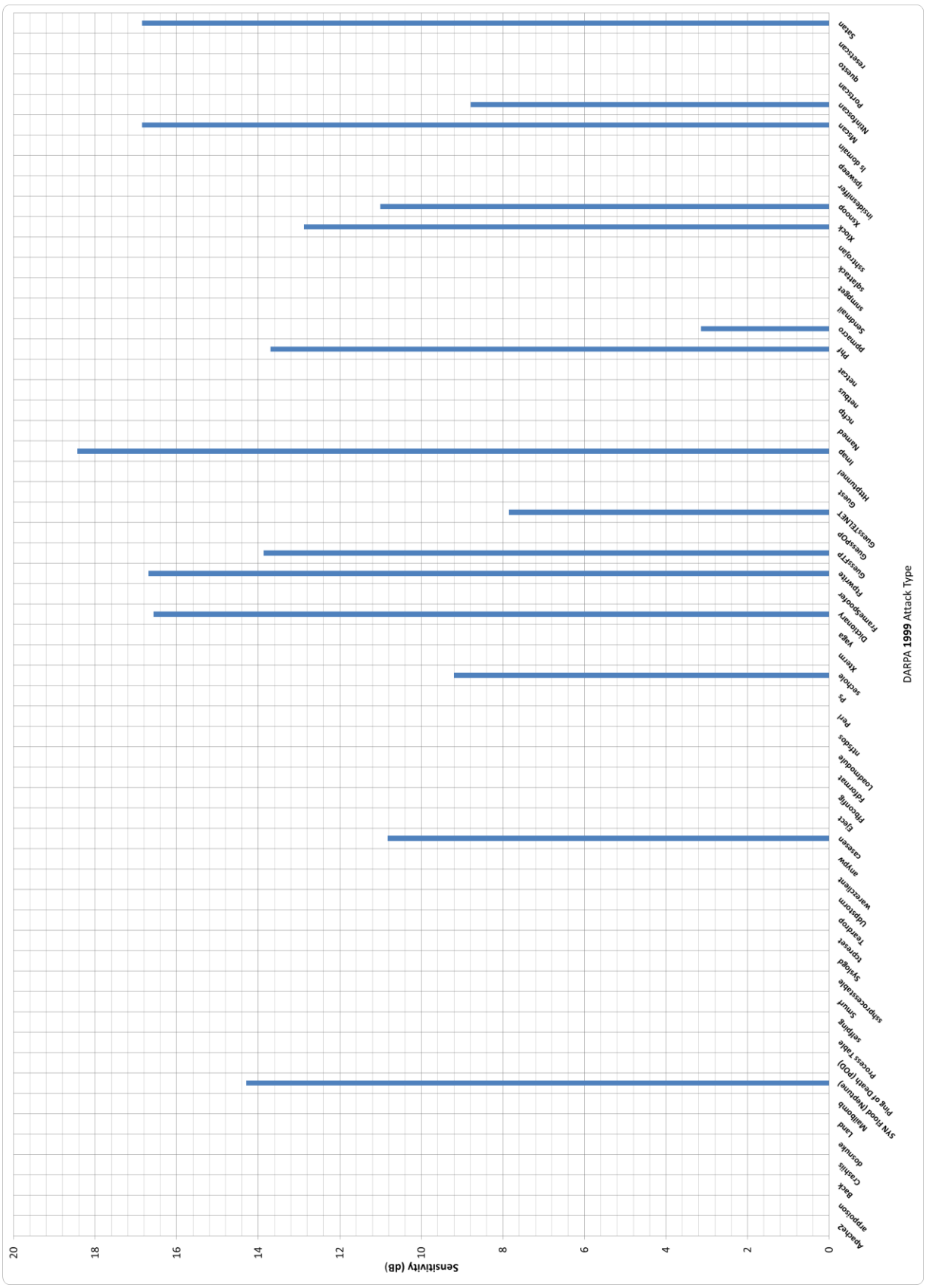
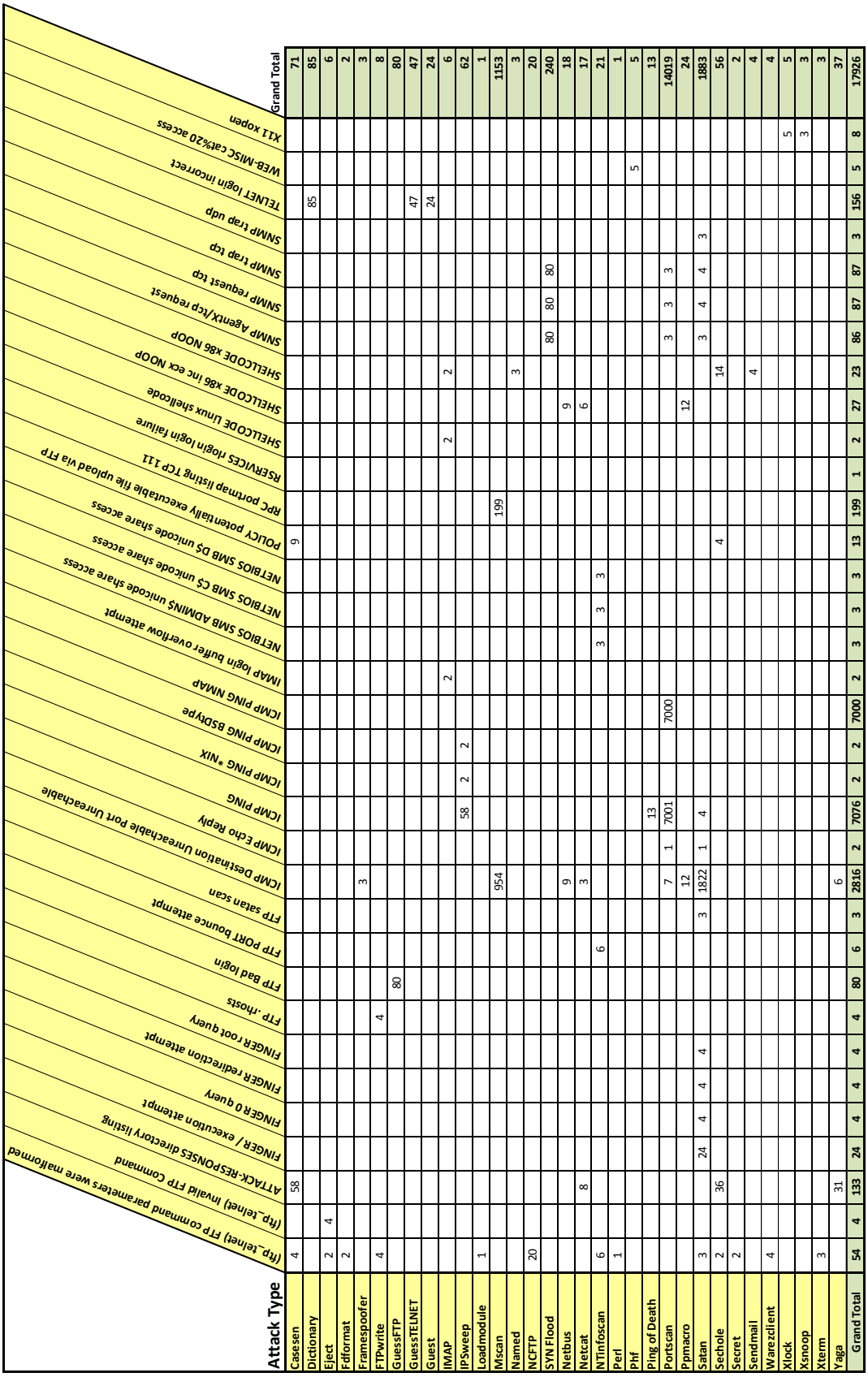


Figure 5-5 Sensitivities of Different Attack Types



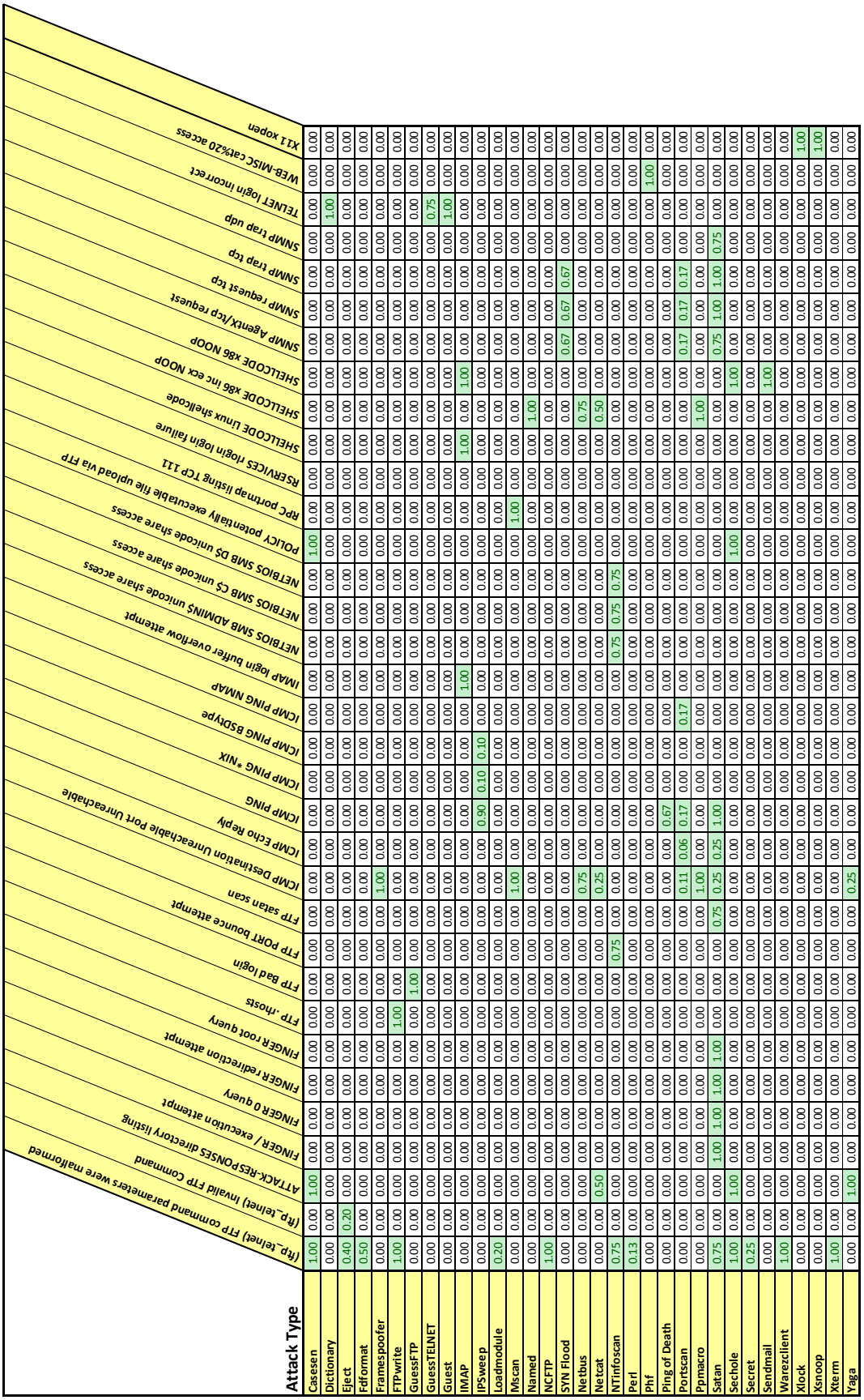
Attack Type	(ftp_telnet) FTP command parameters were malformed	(ftp_telnet) invalid FTP command	ATTACK-RESPONSES directory listing	FINGER / execution attempt	FINGER 0 query	FINGER redirection attempt	FINGER root query	FTP .rhosts	FTP bad login	FTP PORT bounce attempt	FTP satan scan	ICMP Destination Unreachable Port Unreachable	ICMP Echo Reply	ICMP PING	ICMP PING *NIX	ICMP PING BSDType	ICMP PING NMAP	IMAP login buffer overflow attempt	NETBIOS SMB ADMIN$ unicode share access	NETBIOS SMB C$ unicode share access	NETBIOS SMB D$ unicode share access	POLICY potentially executable file upload via FTP	RPC portmap listing TCP 111	RSERVICES rlogin login failure	SHELLCODE Linux shellcode	SHELLCODE x86 inc ecx NOOP	SHELLCODE x86 NOOP	SNMP AgentX/tcp request	SNMP request tcp	SNMP trap tcp	SNMP trap udp	TELNET login incorrect	WEB-MISC cat%20 access	X11 xopen	Grand Total
Casesen	4	58																				9												71
Dictionary																																85		85
Eject	2	4																																6
Fdformat	2																																	2
Framespoofer										3																								3
FTPwrite	4							4																										8
GuessFTP									80																									80
GuessTELNET																															47			47
Guest																															24			24
IMAP																	2									2								6
IPSweep												58	2	2																				62
Loadmodule	1																																	1
Mscan											954											199												1153
Named																										3								3
NCFTP	20																																	20
SYN Flood																												80	80					240
Netbus											9													9										18
Netcat		8									3													6										17
NTinfoscan	6									6								3	3	3														21
Perl	1																																	1
Phf																																	5	5
Ping of Death												13																						13
Portscan											7	7001				7000												3	3	3				14019
Ppmacro											12													12										24
Satan	3		24	4	4	4				3	1822	1	4															3	4	4	3			1883
Sechole	2	36																			4					14								56
Secret	2																																	2
Sendmail																										4								4
Warezclient	4																																	4
Xlock																																5		5
Xsnoop																																3		3
Xterm	3																																	3
Yaga		31									6																							37
Grand Total	54	133	24	4	4	4	4	4	80	6	3	2816	2	7076	2	2	7000	2	3	3	3	13	199	1	2	27	23	86	87	87	3	156	5	8	17926

Table 5-11 Attack Type vs SNORT Signature

By examining Table 5-11 a number of conclusions can be drawn. First, the interaction between signatures and attack types is sparse, with only a small number of non-zero values present. Tantalisingly the number of signatures (34) is comparable with the number of detected attack types (33). However since there is not a one-to-one mapping, the perfect association between signature and attack type is clearly not present. Secondly, some signatures are triggered in multiple attack types, most notably the "(ftp telnet) FTP command parameters were malformed" signature. This implies that the presence of these signatures is not as good at discriminating attack types as other signatures. Thirdly, some attack types trigger multiple signatures and signature types, improving the discrimination between them.

Table 5-12 shows an alternative way of presenting the information from Table 5-11, where the probability of a given signature triggering when an attack type is underway is shown. The non-zero probabilities are highlighted in green to make them easier to locate. Of particular interest in this table is the large number of probability values of 1.0. This may be a factor of the small number of attacks present in each attack type in the DARPA data. Typically there are less than five instances of each attack type and if a given signature is triggered during each attack, then a probability of 1.0 is recorded.

Table 5-13 shows a heat map of the selectivity of the given set of SNORT signatures applied to the attack types present in the DARPA dataset. The colour coding that is applied highlights challenging discriminations in red and shows easier discriminations in green. The leading diagonal is shown in red with a



Attack Type	(ftp_telnet) FTP command parameters were malformed	(ftp_telnet) Invalid FTP command	ATTACK-RESPONSES directory listing	FINGER / execution attempt	FINGER 0 query	FINGER redirection attempt	FINGER root query	FTP .rhosts	FTP bad login	FTP PORT bounce attempt	FTP satan scan	ICMP Destination Unreachable Port Unreachable	ICMP Echo Reply	ICMP PING	ICMP PING *NIX	ICMP PING BSDType	ICMP PING NMAP	IMAP login buffer overflow attempt	NETBIOS SMB ADMIN$ unicode share access	NETBIOS SMB C$ unicode share access	NETBIOS SMB D$ unicode share access	POLICY potentially executable file upload via FTP	RPC portmap listing TCP 111	RSERVICES rlogin login failure	SHELLCODE Linux shellcode	SHELLCODE x86 inc ecx NOOP	SHELLCODE x86 NOOP	SNMP AgentX/tcp request	SNMP request tcp	SNMP trap tcp	SNMP trap udp	TELNET login incorrect	WEB-MISC cat%20 access	X11 xopen
Casesen	1.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Dictionary	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
Eject	0.40	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Fdformat	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Framespoofer	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FTPwrite	1.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GuessFTP	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GuessTELNET	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.75	0.00	0.00
Guest	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
IMAP	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
IPSweep	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.90	0.10	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Loadmodule	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Mscan	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Named	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
NCFTP	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SYN Flood	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.67	0.67	0.67	0.00	0.00	0.00	0.00	0.00
Netbus	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Netcat	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
NTinfoscan	0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.75	0.75	0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Perl	0.13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Phf	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
Ping of Death	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.67	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Portscan	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.11	0.06	0.17	0.00	0.00	0.00	0.17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.17	0.17	0.17	0.00	0.00	0.00	0.00	0.00
Ppmacro	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Satan	0.75	0.00	0.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.75	0.25	0.25	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.75	1.00	1.00	0.75	0.00	0.00	0.00
Sechole	1.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
Secret	0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Sendmail	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
Warezclient	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Xlock	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
Xsnoop	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
Xterm	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Yaga	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 5-12 Probability of Individual Signatures vs Attack Type

number of off-diagonal values also presenting discrimination difficulties. There are a number of cells where discrimination is easier (green) most notably the Satan, Sechole and Casesen attack types.

The largest value in Table 5-13 is 3.52, between the Satan-IMAP attack types. Theoretically, the largest value could be 57.6 if all 3,321 signatures are considered although a maximum of 5.83 would apply if the non-triggering signatures are excluded. This indicates that although there is discrimination highlighted on this heat map, it is significantly lower that the theoretical limit.

To illustrate the use of the selectivity matrix consider again the two attack types of FTPWrite and Xterm, along with the signatures that were triggered during these attacks, as shown in Table 5-9. The probability vector has component entries that are zero except for the "(ftp_telnet) FTP command parameters were malformed" signature (both FTPWrite and Xterm) and "FTP .rhosts" signature (FTPWrite only) which are at unity. Consider three cases for the signatures triggered for a specific connection between two network devices:

- Case 1 - only the "(ftp_telnet) FTP command parameters were malformed" signature is triggered;
- Case 2 – both the "(ftp_telnet) FTP command parameters were malformed" and "FTP .rhosts" signatures are triggered; and
- Case 3 - both the "(ftp_telnet) FTP command parameters were malformed" and "FTP .rhosts" signatures are triggered, along with a third signature, for example "WEB-MISC cat%20 access".

Attack Type	Casesen	Dictionary	Eject	Fdformat	Framespoofer	FTPwrite	GuessFTP	GuessTELNET	Guest	IMAP	IPSweep	Loadmodule	Mscan	Named	NCFTP	SYN Flood	Netbus	Netcat	NTinfoscan	Perl	Phf	Ping of Death	Portscan	Ppmacro	Satan	Sechole	Secret	Sendmail	Warezclient	Xlock	Xsnoop	Xterm	Yaga
Casesen	0.00	2.00	1.55	1.50	2.00	1.73	2.00	1.89	2.00	2.45	1.96	1.62	2.24	2.00	1.41	2.08	2.03	1.60	2.08	1.66	2.00	1.86	1.78	2.24	3.30	1.00	1.60	2.00	1.41	2.00	2.00	1.41	1.44
Dictionary	2.00	0.00	1.10	1.12	1.41	1.73	1.41	0.25	0.00	2.00	1.35	1.02	1.73	1.41	1.41	1.53	1.46	1.25	1.95	1.01	1.41	1.20	1.08	1.73	3.22	2.24	1.03	1.41	1.41	1.41	1.41	1.41	1.44
Eject	1.55	1.10	0.00	0.22	1.10	1.18	1.10	0.87	1.10	1.79	1.01	0.28	1.48	1.10	0.63	1.24	1.15	0.87	1.55	0.34	1.10	0.80	0.60	1.48	3.00	1.84	0.25	1.10	0.63	1.10	1.10	0.63	1.12
Fdformat	1.50	1.12	0.22	0.00	1.12	1.12	1.12	0.90	1.12	1.80	1.04	0.30	1.50	1.12	0.50	1.26	1.17	0.90	1.52	0.38	1.12	0.83	0.64	1.50	2.98	1.80	0.25	1.12	0.50	1.12	1.12	0.50	1.15
Framespoofer	2.00	1.41	1.10	1.12	0.00	1.73	1.41	1.25	1.41	2.00	1.35	1.02	1.00	1.41	1.41	1.53	0.79	1.03	1.95	1.01	1.41	1.20	0.97	1.00	3.14	2.24	1.03	1.41	1.41	1.41	1.41	1.41	1.25
FTPwrite	1.73	1.73	1.18	1.12	1.73	0.00	1.73	1.60	1.73	2.24	1.68	1.28	2.00	1.73	1.00	1.83	1.77	1.60	1.82	1.33	1.73	1.56	1.47	2.00	3.14	2.00	1.25	1.73	1.00	1.73	1.73	1.00	1.75
GuessFTP	2.00	1.41	1.10	1.12	1.41	1.73	0.00	1.25	1.41	2.00	1.35	1.02	1.73	1.41	1.41	1.53	1.46	1.25	1.95	1.01	1.41	1.20	1.08	1.73	3.22	2.24	1.03	1.41	1.41	1.41	1.41	1.41	1.44
GuessTELNET	1.89	0.25	0.87	0.90	1.25	1.60	1.25	0.00	0.25	1.89	1.18	0.78	1.60	1.25	1.25	1.38	1.30	1.06	1.84	0.76	1.25	1.00	0.85	1.60	3.15	2.14	0.79	1.25	1.25	1.25	1.25	1.25	1.27
Guest	2.00	0.00	1.10	1.12	1.41	1.73	1.41	0.25	0.00	2.00	1.35	1.02	1.73	1.41	1.41	1.53	1.46	1.25	1.95	1.01	1.41	1.20	1.08	1.73	3.22	2.24	1.03	1.41	1.41	1.41	1.41	1.41	1.44
IMAP	2.45	2.00	1.79	1.80	2.00	2.24	2.00	1.89	2.00	0.00	1.96	1.74	2.24	2.00	2.00	2.08	2.03	1.89	2.41	1.74	2.00	1.86	1.78	2.24	3.52	2.24	1.75	2.00	2.00	2.00	2.00	2.00	2.02
IPSweep	1.96	1.35	1.01	1.04	1.35	1.68	1.35	1.18	1.35	1.96	0.00	0.93	1.68	1.35	1.35	1.48	1.40	1.18	1.91	0.92	1.35	0.27	0.83	1.68	2.90	2.20	0.94	1.35	1.35	1.35	1.35	1.35	1.38
Loadmodule	1.62	1.02	0.28	0.30	1.02	1.28	1.02	0.78	1.02	1.74	0.93	0.00	1.43	1.02	0.80	1.18	1.08	0.78	1.60	0.08	1.02	0.70	0.45	1.43	3.02	1.91	0.05	1.02	0.80	1.02	1.02	0.80	1.05
Mscan	2.24	1.73	1.48	1.50	1.00	2.00	1.73	1.60	1.73	2.24	1.68	1.43	0.00	1.73	1.73	1.83	1.27	1.44	2.19	1.42	1.73	1.56	1.39	1.41	3.30	2.45	1.44	1.73	1.73	1.73	1.73	1.73	1.60
Named	2.00	1.41	1.10	1.12	1.41	1.73	1.41	1.25	1.41	2.00	1.35	1.02	1.73	0.00	1.41	1.53	0.79	0.75	1.95	1.01	1.41	1.20	1.08	1.00	3.22	2.24	1.03	1.41	1.41	1.41	1.41	1.41	1.44
NCFTP	1.41	1.41	0.63	0.50	1.41	1.00	1.41	1.25	1.41	2.00	1.35	0.80	1.73	1.41	0.00	1.53	1.46	1.25	1.52	0.88	1.41	1.20	1.08	1.73	2.98	1.73	0.75	1.41	0.00	1.41	1.41	0.00	1.44
SYN Flood	2.08	1.53	1.24	1.26	1.53	1.83	1.53	1.38	1.53	2.08	1.48	1.18	1.83	1.53	1.53	0.00	1.57	1.38	2.04	1.17	1.53	1.34	0.91	1.83	2.65	2.31	1.19	1.53	1.53	1.53	1.53	1.53	1.55
Netbus	2.03	1.46	1.15	1.17	0.79	1.77	1.46	1.30	1.46	2.03	1.40	1.08	1.27	0.79	1.46	1.57	0.00	0.75	1.98	1.07	1.46	1.25	1.06	0.35	3.18	2.26	1.09	1.46	1.46	1.46	1.46	1.46	1.35
Netcat	1.60	1.25	0.87	0.90	1.03	1.60	1.25	1.06	1.25	1.89	1.18	0.78	1.44	0.75	1.25	1.38	0.75	0.00	1.84	0.76	1.25	1.00	0.82	1.03	3.13	1.89	0.79	1.25	1.25	1.25	1.25	1.25	0.71
NTinfoscan	2.08	1.95	1.55	1.52	1.95	1.82	1.95	1.84	1.95	2.41	1.91	1.60	2.19	1.95	1.52	2.04	1.98	1.84	0.00	1.63	1.95	1.80	1.72	2.19	3.33	2.30	1.58	1.95	1.52	1.95	1.95	1.52	1.97
Perl	1.66	1.01	0.34	0.38	1.01	1.33	1.01	0.76	1.01	1.74	0.92	0.08	1.42	1.01	0.88	1.17	1.07	0.76	1.63	0.00	1.01	0.68	0.42	1.42	3.03	1.94	0.13	1.01	0.88	1.01	1.01	0.88	1.04
Phf	2.00	1.41	1.10	1.12	1.41	1.73	1.41	1.25	1.41	2.00	1.35	1.02	1.73	1.41	1.41	1.53	1.46	1.25	1.95	1.01	0.00	1.20	1.08	1.73	3.22	2.24	1.03	1.41	1.41	1.41	1.41	1.41	1.44
Ping of Death	1.86	1.20	0.80	0.83	1.20	1.56	1.20	1.00	1.20	1.86	0.27	0.70	1.56	1.20	1.20	1.34	1.25	1.00	1.80	0.68	1.20	0.00	0.62	1.56	2.91	2.11	0.71	1.20	1.20	1.20	1.20	1.20	1.23
Portscan	1.78	1.08	0.60	0.64	0.97	1.47	1.08	0.85	1.08	1.78	0.83	0.45	1.39	1.08	1.08	0.91	1.06	0.82	1.72	0.42	1.08	0.62	0.00	1.39	2.86	2.04	0.47	1.08	1.08	1.08	1.08	1.08	1.08
Ppmacro	2.24	1.73	1.48	1.50	1.00	2.00	1.73	1.60	1.73	2.24	1.68	1.43	1.41	1.00	1.73	1.83	0.35	1.03	2.19	1.42	1.73	1.56	1.39	0.00	3.30	2.45	1.44	1.73	1.73	1.73	1.73	1.73	1.60
Satan	3.30	3.22	3.00	2.98	3.14	3.14	3.22	3.15	3.22	3.52	2.90	3.02	3.30	3.22	2.65	2.65	3.18	3.13	3.33	3.03	3.22	2.91	2.86	3.30	0.00	3.45	3.01	3.22	3.98	3.22	3.22	3.22	3.21
Sechole	1.00	2.24	1.84	1.80	2.24	2.00	2.24	2.14	2.24	2.24	2.20	1.91	2.45	2.24	1.73	2.31	2.26	1.89	2.30	1.94	2.24	2.11	2.04	2.45	3.45	0.00	1.89	1.73	1.73	2.24	2.24	1.73	1.75
Secret	1.60	1.03	0.25	0.25	1.03	1.25	1.03	0.79	1.03	1.75	0.94	0.05	1.44	1.03	0.75	1.19	1.09	0.79	1.58	0.13	1.03	0.71	0.47	1.44	3.01	1.89	0.00	1.03	0.75	1.03	1.03	0.75	1.06
Sendmail	2.00	1.41	1.10	1.12	1.41	1.73	1.41	1.25	1.41	1.41	1.35	1.02	1.73	1.41	1.41	1.53	1.46	1.25	1.95	1.01	1.41	1.20	1.08	1.73	3.22	1.73	1.03	0.00	1.41	1.41	1.41	1.41	1.44
Warezclient	1.41	1.41	0.63	0.50	1.41	1.00	1.41	1.25	1.41	2.00	1.35	0.80	1.73	1.41	0.00	1.53	1.46	1.25	1.52	0.88	1.41	1.20	1.08	1.73	2.98	1.73	0.75	1.41	0.00	1.41	1.41	0.00	1.44
Xlock	2.00	1.41	1.10	1.12	1.41	1.73	1.41	1.25	1.41	2.00	1.35	1.02	1.73	1.41	1.41	1.53	1.46	1.25	1.95	1.01	1.41	1.20	1.08	1.73	3.22	2.24	1.03	1.41	1.41	0.00	0.00	1.41	1.44
Xsnoop	2.00	1.41	1.10	1.12	1.41	1.73	1.41	1.25	1.41	2.00	1.35	1.02	1.73	1.41	1.41	1.53	1.46	1.25	1.95	1.01	1.41	1.20	1.08	1.73	3.22	2.24	1.03	1.41	1.41	0.00	0.00	1.41	1.44
Xterm	1.41	1.41	0.63	0.50	1.41	1.00	1.41	1.25	1.41	2.00	1.35	0.80	1.73	1.41	0.00	1.53	1.46	1.25	1.52	0.88	1.41	1.20	1.08	1.73	2.98	1.73	0.75	1.41	0.00	1.41	1.41	0.00	1.44
Yaga	1.44	1.44	1.12	1.15	1.25	1.75	1.44	1.27	1.44	2.02	1.38	1.05	1.60	1.44	1.44	1.55	1.35	0.71	1.97	1.04	1.44	1.23	1.08	1.60	3.21	1.75	1.06	1.44	1.44	1.44	1.44	1.44	0.00

Table 5-13 Selectivity Heatmap for Different Attack Types

For each case, the distance, as defined in Equation 4-9, between the measured probability vector and the row vectors representing each attack type shown in Table 5-12 is calculated and shown in Table 5-14 below.

Attack Type	Distance Case 1	Distance Case 2	Distance Case 3
Casesen	1.414	1.732	2
Dictionary	1.414	1.732	2
Eject	0.632	1.183	1.549
Fdformat	0.5	1.118	1.5
Framespoofer	1.414	1.732	2
FTPwrite	1	0	1
GuessFTP	1.414	1.732	2
GuessTELNET	1.25	1.601	1.887
Guest	1.414	1.732	2
IMAP	2	2.236	2.449
IPSweep	1.353	1.682	1.957
Loadmodule	0.8	1.281	1.625
Mscan	1.732	2	2.236
Named	1.414	1.732	2
NCFTP	0	1	1.414
SYN Flood	1.532	1.829	2.085
Netbus	1.458	1.768	2.031
Netcat	1.25	1.601	1.887
NTinfoscan	1.521	1.82	2.077
Perl	0.875	1.329	1.663
Phf	1.414	1.732	1.414
Ping of Death	1.202	1.564	1.856
Portscan	1.077	1.47	1.778
Ppmacro	1.732	2	2.236
Satan	2.979	3.142	3.298
Sechole	1.732	2	2.236

Attack Type	Distance Case 1	Distance Case 2	Distance Case 3
Secret	0.75	1.25	1.601
Sendmail	1.414	1.732	2
Warezclient	0	1	1.414
Xlock	1.414	1.732	2
Xsnoop	1.414	1.732	2
Xterm	0	1	1.414
Yaga	1.436	1.75	2.016

Table 5-14 Examples of the Use of Selectivity

The smaller the distance the nearer the measured probability vector is to a given attack type. For Case 1 there are three attacks that make a perfect match, that is, NCFTP, Warezclient and Xterm. It is not possible to discriminate between these attack types given the probability vector measured for Case 1.

For Case 2, there is only one perfect match, namely FTPWrite. All other attack types are at least a distance of 1.0 away and therefore this represents a confident selection of the FTPWrite attack for the measured probability vector. The nearest alternative attack types are still NCFTP, Warezclient and Xterm, but the presence of an additional measurement in the Case 2 probability vector improves the discrimination over Case 1.

For Case 3, the smallest distance is still the FTPWrite attack type. However the distance is not zero indicating that this is not a perfect match. Alternative attacks are now only a distance of 0.414 away and there are now four nearest neighbours, namely NCFTP, Warezclient, Xterm and Phf.

These cases illustrate some important points about the use of selectivity. There is significance to the measured distance, with zero indicating perfect match, but

not necessarily perfect discrimination. Additional measurements can improve the discrimination but also may no longer achieve perfect matching to an attack type. Also, not all additional measurements will improve discrimination, but they may reduce the distance to alternative attack types.

In order to apply this approach, after measuring the distance a further level of processing is necessary to either declare a specific intrusion event is underway or to provide information to other security devices on the network. Two obvious processing approaches are:

- Intrusion Declaration – in which decision thresholds are applied to the shortest distance and to the separation between the shortest distance and the distance to alternative attack types, in order to declare an intrusion. The shortest distance would need to be below the first threshold and the separation of alternative attacks above the second threshold to declare an intrusion; and
- Probabilistic or Fuzzy Approach – in which no decision is made by the NIS. Instead a vector of the distance to all attack types is output to combine with information from other security devices in a hierarchical approach. In some implementations it may be better to output the vector as a measure of probability rather than distance, for inclusion in expert systems or evidential reasoning.

The investigation and optimisation of this additional processing will be the subject of further research.

5.5. Discussion

The results shown in the previous section illustrate the use of sensitivity and selectivity in the performance assessment of NIS. The results show that sensitivity is poor, with only 15 of the 58 attack types present showing a non-zero value, despite 33 attacks containing SNORT alerts. The selectivity of the evaluated set of SNORT signatures is also poor indicating that accurate discrimination between the different attack types is difficult.

Although these results are poor they could be improved through tuning the selection of SNORT signatures so that only those with the most advantageous impact on sensitivity and selectivity are included. Whilst this appears to be the right approach, particularly with the observation that only 34 of the 3,321 SNORT signatures were triggered for valid intrusions, the process of achieving this optimisation is not straight forward. It is proposed that this is achieved in the following way:

- Security Policy Mapping – with each individual requirement in the network security policy represented by at least one signature, but ideally more. This will enable policy violations to be the focus of signature selection rather than intrusion mechanisms;
- Standardised Policy Violations – in which a standard set of intrusions are defined to highlight all known ways in which individual policy violations can be achieved. The use of standards will allow comparison between NIS operating on different live networks; and
- Automated Analysis – in which the set of standardised attacks are applied over a live network with automated sensitivity and selectivity

measurements produced.

It is likely that such an approach will take considerable time to build up false alarm statistics, as well and sensitivity and selectivity measurements. This will be the topic of future research.

5.6. Summary and Conclusions

This chapter has demonstrated the application of the performance metrics sensitivity and selectivity to the simulated attacks present in the DARPA 1999 dataset, using SNORT as an exemplar signature-based NIS. The achieved performance for both metrics was poor with an inability to detect many of the attacks, as well as discriminating between individual attacks.

The utility of sensitivity as a performance measure has been demonstrated. In particular sensitivity can be used to distinguish between fortuitous detection by noisy signatures which produce a high false alarm rate, and detection by signatures responding to the attack specifics. At best noisy signatures would be given a lower priority by network support staff and at worse noisy signatures would be disabled. The use of sensitivity provides an alternative approach in which the significance of a given alert is determined based on the other signatures triggered for the same connection.

The sensitivity metric replaces the usual four measures of true positive, true negative, false positive and false negative rates with a single number. It represents the fundamental performance of a NIS as without detection of an attack type the discrimination implied by the selectivity metric cannot occur. Only nine of the intrusion event types present in the DARPA dataset were

detected with a sensitivity greater than 12dB.

Selectivity measurements have shown that it is possible to distinguish between a small number of attack types. However the highest value of selectivity was 3.52, which is significantly lower than the theoretical limit of 5.83, indicating that there is significant room for improvement.

In order to use the taxonomy described in Chapter 3 improvements in performance is necessary. Sensitivity will need to be improved to ensure that more attack types are detected with a high confidence. Selectivity will be needed for recognition, identification and confirmation to be successful.

A summary of the conclusions of this thesis will now be presented.

CHAPTER 6

SUMMARY AND CONCLUSIONS

6. Summary and Conclusions

This chapter provides a summary of the research activities that have been undertaken along with their principal conclusions. All the objectives of this research have been achieved in full, as follows:

a) Objective 1 - Review the techniques and performance measures that have been applied to intrusion systems, to identify the most promising techniques for further evaluation; – A literature review has been conducted and is reported in Chapter 2;

b) Objective 2 – Re-evaluate the meaning of "detection" in the context of NIS – Detection has been defined in Chapter 3, along with other high-level intrusion system functions;

c) Objective 3 - Assess the application of detection theory to NIS and propose metrics that can be used to characterise their performance – Chapter 4 describes a systems engineering approach to NIS and defines two performance metrics new to intrusion systems, namely sensitivity and selectivity; and

d) Objective 4 - Demonstrate experimentally the use of the performance metrics and the potential for false alarm rejection using a representative NIS and practical data – Chapter 5 describes the use of SNORT against the DARPA 1999 inside network dataset, with practical sensitivity and selectivity measurements being undertaken.

6.1. Summary of Research Activities

A programme of research has been undertaken to determine improved metrics for comparing intrusion systems, using techniques developed from other detection-based technologies, such as used in radar or sonar.

The research commenced with a literature review. As intrusion detection has been an active research topic for over 30 years considerable published work is available. A detailed systems and techniques assessment of other research was made with particular emphasis given to performance evaluation metrics and their limitations. The survey identified that a wide range of data processing techniques had been assessed and a large number of different intrusion systems developed for research and commercial purposes. However the problem of poor performance persists and the difficulty of comparing intrusion techniques and systems was highlighted.

A taxonomy of intrusion systems was then developed as a basis for a more precise definition of "detection". Key NIS discrimination technologies identified in the literature were mapped onto a graphical representation of the taxonomy to illustrate how meaningful comparison could and could not be made. The link between an intrusion event and the network security policy was established.

A systems-level assessment of the issues facing NIS was undertaken to identify key considerations in defining performance. The reasons for deployment and the characteristics of an ideal system were developed and current challenges defined. Two metrics known as sensitivity and selectivity were proposed to measure the detection and discrimination performance respectively. These

metrics were defined mathematically and guidance on their interpretation in intrusion detection was developed.

Finally, an experimental evaluation of NIS was undertaken using SNORT and the DARPA 1999 dataset of network frames recorded on the inside of the simulated network. A new set of truth data was developed to enable the correlation of SNORT intrusion alerts with DARPA attack simulations. A manual assignment of SNORT alerts to specific attacks or to false alarms was made. From this assignment measurements of sensitivity and selectivity were undertaken illustrating the application of these metrics. The results indicated that SNORT, with a baseline set of signatures from 2010 has poor sensitivity and selectivity for attacks in the DARPA 1999 dataset.

6.2. Research Achievements

As well as achieving the research objectives a number of original contributions to the corpus of knowledge regarding network intrusion systems has been made including:

- The literature survey identified inconsistencies in the terminology used for intrusion systems with researchers rarely defining what is meant by "detection" with some describing the process as "intrusion recognition". Difficulties in comparing practical measured performances were also identified, in particular the inconsistent use of receiver operating characteristic curves between different research groups;
- A novel taxonomy has been developed based of the output of an intrusion system and the data scale over which it operates. It has been demonstrated that this taxonomy is ideally suited to comparing intrusion

systems on an equitable basis. High-level functions have been defined for intrusion systems, based on their required outputs. The acronym NIS has been proposed to generalise intrusion from detection to recognition and identification;

- A new relationship between network security policy and the definition of an intrusion has been developed. An intrusion has been defined as an event that breaches the network security policy. It is proposed that each clause or group of clauses in the policy are used to select at least one signature for misuse-based intrusion systems. The interrelation between network security policy and implied threshold setting in a classical detection system has been described;
- The reasons for the deployment of an intrusion system have been developed along with the characteristics of an ideal system. A model of an first generation ideal passive system has been described and the connection to other design patterns has been established;
- The basic properties of a misuse-based intrusion system have been developed and a fundamental problem identified, namely the non-unique mapping to intrusion-like or non-intrusion-like for some network frames. This problem limits the achievable performance for practical misuse-based intrusion systems;
- The observation that current passive intrusion systems are inefficient has been made. Intrusion systems spend much of their processing effort processing non-intrusion data. The implications of this for the development of advanced data processing techniques is described;

- Two performance metrics, sensitivity and selectivity, new to intrusion systems have been defined mathematically and their properties developed. Their application to intrusion systems has been described;
- Sensitivity and selectivity has been measured for the SNORT intrusion system processing data from the DARPA 1999 network simulation. The utility of sensitivity to describe performance in terms of a single parameter rather than the four performance measures has been established. The use of selectivity in undertaking high-level intrusion functions, as described in the novel taxonomy described above, has been described; and
- New limitations of the DARPA 1999 database for intrusion system evaluation have been identified. Poor time synchronisation between the inside and outside datasets limit their simultaneous use without different truth data for each.

The use and definition of sensitivity and selectivity was based on detection theory as used within other technologies, such as radar and sonar. During the course of this research it was observed that other analogies could be made with these technologies, specifically allowing the NIS to stimulate deliberately the network to gather further information to improve performance. This is reported in Appendix D and is given the term Aggressive Network Intrusion System (AgNIS) to differentiate the approach from Active Intrusion Detection. Although this approach is not new to NIS, a systematic evaluation of the methods for integration did not appear to have been addressed by other researchers.

6.3. Research Limitations

As with all research the depth with which individual topics could be investigated is limited by the available effort and the need to achieve all the objectives of the work. In this research three significant limitations have been identified, as follows:

- Performance Metrics – The impact of the assumption of a Gaussian probability density function has not been established precisely. Alternative parameterisations should be investigated, including using measured density functions, to quantify the impact. In addition, the sensitivity and selectivity metrics should be extended to high-level intrusion functions, such as recognition and identification;
- Use of the DARPA 1999 Dataset – This dataset has a number of well-known issues with new limitations being identified in this research. It is now 14 years old and not representative of modern networks. The goal of the practical phase was not to optimise performance, if this were not the case the DARPA dataset would have been unsuitable. The demonstration of the use of the new taxonomy as well as the sensitivity and selectivity metrics on a live network is required; and
- Exploitation of the new metrics – The retrospective measurement of sensitivity and selectivity from the published results of other research teams has not been attempted, mainly due to the difficulty of mapping their results to this new measurement framework. Although a ROC curve can be used to estimate sensitivity, through curve fitting, the differences between the measurement methodologies used by different research

groups would still make this a difficult task.

In order to evaluate further the limitations of this work it was decided to gauge the view of other professionals by obtaining independent review of the body of this thesis. This is reported fully in Appendix F. No further limitations were identified and the taxonomy and new metrics were thought to be of value by a consulting security architect.

6.4. Further Work

Further work can be considered for both the experimental and systems aspects of this research. Throughout this thesis opportunities for further research have already been described. In this chapter a new research topic is proposed based on addressing the limitations of the current research, as described in the previous section.

It is proposed that an NIS demonstrator is constructed and operated over a live network to establish the practical use of these metrics in real-time systems. The calculation of performance metrics should be automated. In order for this to occur a standardised set of attack types will need to be developed along with methods for automatically assigning signature alerts to simulated attacks. Full network recording at the NIS will be needed to confirm that any false alarms that are due to system limitations and not genuine, unapproved intrusion behaviour against the intrusion system. It is anticipated that real-time performance could be achieved by assuming that no genuine intrusion behaviour is present, with offline correction as this assumption is verified. This demonstrator would enable benchmarking of intrusion systems.

The demonstrator should record all statistics so that the practical implications of the Gaussian probability density function assumption can be assessed. This should be supplemented by a theoretical assessment using both alternative density functions and the Tchebycheff inequality.

Strategies for the selection of signatures will need to be developed and evaluated using the performance metrics. The practicality of mapping signatures to different network security policies will be determined. The relationship between a given set of signatures and the achievable sensitivity and sensitivity should be examined and quantified.

Finally, the range of performance metrics should be extended to cover high-level intrusion functionality identified by the taxonomy described in Chapter 3. Such an experimental programme as proposed in this chapter could be the basis for developing high-level intrusion functionality, becoming a component of a new form of distributed intelligent security paradigm. New architectures or data processing techniques, such as AgNIS, could be evaluated in real-time and their performance contrasted with alternative approaches.

6.5. The Future for Network Intrusion Systems

Despite over 30 years of active research, the performance achieved by network intrusion systems remains inadequate. False alarms, both positive and negative are at unacceptable levels. False positives require significant system administrator effort to investigate whilst false negatives are more insidious giving administrators the false perception that their networks are secure.

The programme of research described here has created a more precise definition of detection and high-level intrusion functions such that performance comparison between systems and data processing algorithms can confidently be made. The use of sensitivity and selectivity as basic performance metrics, along with the future development of standardised attacks, should highlight the most promising data processing techniques and systems approaches. The goal should be for future research to be published in terms of these standard metrics on live data, rather than the current trend of using synthesised datasets.

Alternative systems approaches to intruion will be required, such as AgNIS, to address improved sensitivity for attacks which have only a minor impact on network traffic. Techniques to improve selectivity will also be required, as the functionality with NIS moves towards recognition and identification systems.

Finally, the demand for high performance intrusion systems is unlikely to diminish. Although a defence in depth strategy for security architectures, coupled with improving performance in other network security elements will limit the potential for damage from intruders, it is likely to remain a key requirement that the status of a network as intrusion-free will need to be confirmed continuously. In order for this to be an overall benefit rather than a resource burden, high sensitivity and selectivity is required across the complete range of attack types implied by the organisations network security policy.

List of References

1). Abad, C., J. Taylor, et al. (2003). "Log correlation for intrusion detection: a proof of concept", Computer Security Applications Conference, 2003. Proceedings. 19th Annual.

2). Abouzakhar, N. S. and G. A. Manson (2004). "Evaluation of intelligent intrusion detection models", International Journal of Digital Evidence **3**(1).

3). Abramowitz, M. and I. A. Stegun (1964). "Handbook of Mathematical Functions", Dover Publications.

4). Afzaal, M., C. Di Sarno, et al. (2012). "A Resilient Architecture for Forensic Storage of Events in Critical Infrastructures", High-Assurance Systems Engineering (HASE), 2012 IEEE 14th International Symposium on.

5). Agah, A., S. K. Das, et al. (2004). "Intrusion detection in sensor networks: A non-cooperative game approach", Proceedings of the Third IEEE International Symposium on Network Computing and Applications (NCA '04).

6). Ahsan, K. (2002). "Covert channel analysis and data hiding in TCP/IP", Masters of Applied Science, University of Toronto.

7). Aickelin, U., P. Bentley, et al. (2003). "Danger theory: The link between AIS and IDS?", Artificial Immune Systems, Proceedings. Berlin, SPRINGER-VERLAG BERLIN. **2787:** 147-155.

8). Alessandri, D., C. Cachin, et al. (2001). "Towards a taxonomy of intrusion detection systems and attacks", IBM Research, Zurich Research Laboratory.

9). Ali, H. A. (2001). "A new model for monitoring intrusion based on Petri Nets", Information Management and Computer Security **9**(4): 175-182.

10). Allen, J., A. Christie, et al. (1999). "State of the practice of intrusion detection technologies", Carnegie Mellon, Software Engineering Institute.

11). Allen, W. H. and G. A. Marin (2003). "On the self-similarity of synthetic traffic for the evaluation of intrusion detection systems", Applications and the Internet, 2003. Proceedings. 2003 Symposium on.

12). Almasizadeh, J. and M. A. Azgomi (2013). "A Stochastic Model of Attack Process for the Evaluation of Security Metrics", Computer Networks.

13). Alpcan, T. and T. Basar (2003). "A game theoretic approach to decision and analysis in network intrusion detection", 42nd IEEE Conference on Decision and Control, Maui.

14). Ambwani, T. (2003). "Multi class support vector machine implementation to intrusion detection", Proceedings of the International Joint Conference on Neural Networks 2003, Vols 1-4. New York, I E E E**:** 2300-2305.

15). Amoroso, E. G. (1998). "Intrusion Detection : An introduction to Internet surveillance, correlation, traps, race back, and response", Sparta, N.J., Intrusion.Net Books.

16). Anderson, D., T. Frivold, et al. (1995). "Next-generation Intrusion Detection Expert System (NIDES) : a summary", Menlo Park, Calif., SRI International Computer Science Laboratory.

17). Anderson, J. (1980). "Computer security, threat monitoring and surveillance". Fort Washington PA, James P Anderson Co.

18). Andhare, M. A. and A. B. Patil (2012). "Denial-of-Service Attack Detection Using Genetic-Based Algorithm", reproduction **2**(2).

19). Apap, R., A. Honig, et al. (2002). "Detecting malicious software by monitoring anomalous windows registry accesses", Recent Advances in Intrusion Detection, Proceedings. Berlin, SPRINGER-VERLAG BERLIN. **2516:** 36-53.

20). Arkin, O. (1999). "Network scanning techniques", Publicom Communications Solutions.

21). AsSadhan, B. A. (2009). "Network Traffic Analysis Through Statistical Signal Processing Methods", Carnegie Mellon University.

22). Avallone, S., D. Emma, et al. (2004). "A Practical Demonstration of Network TrafficGeneration", Proceedings of the Eighth IASTED International Conference Internet and Multimedia Systems and Applications Hawaii.

23). Axelsson, S. (1999a). "Base-rate fallacy and its implications for the difficulty of intrusion detection", Proceedings of the 1999 6th ACM Conference on Computer and Communications Security (ACM CCS), Nov 2-Nov 4 1999, Singapore, Singapore, ACM, New York, NY, USA.

24). Axelsson, S. (1999b). "Research in intrusion detection systems: A survey". Goteborg, Chalmers University of Technology.

25). Axelsson, S. (2000a). "Intrusion detection systems: A survey and taxonomy", Department of Computer Engineering, Chalmers University.

26). Axelsson, S. (2000b). "A preliminary attempt to apply detection and estimation theory to intrusion detection", Chalmers University of Technology.

27). Bace, R. and P. Mell (2001). "Intrusion detection systems", NIST Special Publication on Intrusion Detection System.

28). Bahaa-Eldin, A. M. (2011). "Time series analysis based models for network abnormal traffic detection", Computer Engineering & Systems (ICCES), 2011 International Conference on.

29). Balasubramaniyan, J. S., J. O. Garcia-Fernandez, et al. (1998). "An architecture for intrusion detection using autonomous agents", 14th IEEE Computer Security Applications Conference.

30). Barford, P., J. Kline, et al. (2002a). "A signal analysis of network traffic anomalies", Proceedings of the 2nd Internet Measurement Workshop (IMW 2002), Nov 6-8 2002, Marseille, France, Association for Computing Machinery.

31). Barford, P., J. Kline, et al. (2002b). "A signal analysis of network traffic anomalies", ACM.

32). Bass, T. (2000). "Intrusion detection systems and multisensor data fusion", Communications of the ACM **43**(4): 99-105.

33). Bayuk, J. (2011). "Cloud security metrics", System of Systems Engineering (SoSE), 2011 6th International Conference on.

34). Bayuk, J. and A. Mostashari (2013). "Measuring systems security", Systems Engineering **16**(1).

35). Beghdad, R. (2009). "Modelling intrusion detection as an allocation problem", Pattern Recognition Letters **30**(8): 774-779.

36). Behera, S. R. (2001). "Towards the automatic generation of mobile agents for intrusion detection system", Masters, Iowa State University.

37). Bolzoni, D. and S. Etalle (2006). "APHRODITE: An anomaly-based architecture for false positive reduction", Arxiv preprint cs/0604026.

38). Bonelli, N., A. Di Pietro, et al. (2012). "Flexible High Performance Traffic Generation on Commodity Multi–core Platforms", Traffic Monitoring and Analysis, Springer**:** 157-170.

39). Bonelli, N., S. Giordano, et al. (2005). "BRUTE: A high performance and extensible traffic generator", Proc. of SPECTS.

40). Borotschnig, H., L. Paletta, et al. (1999). "A Comparison of probabilistic, possibilistic and evidence theoretic fusion schemes for active object recognition.", Computing **62**: 293-319.

41). Botha, M. and R. v. Solms (2004). "Utilizing neural networks for effective intrusion detection", Information Securty South Africa, ISSA'04.

42). Botta, A., A. Dainotti, et al. (2012). "A tool for the generation of realistic network workload for emerging networking scenarios", Computer Networks **56**: 3531-3547.

43). Botta, A., A. Dainotti, et al. (2010). "Do you trust your software-based traffic generator?", Communications Magazine, IEEE **48**(9): 158-165.

44). Boughaci, D., M. L. Herkat, et al. (2012). "A specific fuzzy genetic Algorithm for intrusion detection", Proceedings of ICCIT.

45). Brescia, U. o. (2009). "UNIBS: Data Sharing", 2013, from www.ing.unibs.it/ntw/tools/traces/.

46). Bridges, S. M. and R. B. Vaughn (2000). "Fuzzy data mining and genetic algorithms applied to intrusion detection", Proceedings of the 12th Annual Canadian Information Technology Security Symposium.

47). Buchheim, T., M. Erlinger, et al. (2001). "Implementing the intrusion detection exchange protocol", 17th Annual Computer Security Applications Conference, Proceedings. Los Alamitos, IEEE COMPUTER SOC**:** 32-41.

48). Buennemeyer, T. K., F. Munshi, et al. (2007). "Battery-sensing intrusion protection for wireless handheld computers using a dynamic threshold calculation algorithm for attack detection", IEEE.

49). Burges, C. (1998). "A tutorial on support vector machines for pattern recognition", Data Mining and Knowledge Discovery **2**: 121-167.

50). Caswell, B., J. Beale, et al. (2003). "Snort 2.0 Intrusion Detection", Syngress Publishing Inc.

51). Cha, B., B. Vaidya, et al. (2005). "Anomaly intrusion detection for system call using the soundex algorithm and neural networks", Computers and Communications, 2005. ISCC 2005. Proceedings. 10th IEEE Symposium on.

52). Chan, A. P. F., W. W. Y. Ng, et al. (2004). "Refinement of rule-based intrusion detection system for denial of service attacks by support vector machine", Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on.

53). Chen, T. M. and V. Venkataramanan (2005). "Dempster-Shafer theory for intrusion detection in ad hoc networks", Internet Computing, IEEE **9**(6): 35-41.

54). Cheng-Yuan, H., L. Ying-Dar, et al. (2012). "False positives and negatives from real traffic with intrusion detection/prevention systems". International Conference on Advancesments in Information Technology (AIT 2012). Hong Kong.

55). Cheng-Yuan, H., L. Yuan-Cheng, et al. (2012). "Statistical analysis of false positives and false negatives from real traffic with intrusion detection/prevention systems", Communications Magazine, IEEE **50**(3): 146-154.

56). Cheng, T., Y. Lin, et al. (2012). "Evasion Techniques: Sneaking through Your Intrusion Detection/Prevention Systems", IEEE Communications Surveys & Tutorials **14**(4): 1011-1020.

57). Chinchani, R., S. Upadhyaya, et al. (2002). "Towards the scalable implementation of a user level anomaly detection system", 2002 Milcom Proceedings, Vols 1 and 2 - Global Information Grid - Enabling Transformation through 21st Century Communications. New York, I E E E**:** 1503-1508.

58). Chrun, D., M. Cukier, et al. (2008). "On the Use of Security Metrics Based on Intrusion Prevention System Event Data: An Empirical Analysis", High Assurance Systems Engineering Symposium, 2008. HASE 2008. 11th IEEE.

59). COAST (2012). "Intrusion Detection". http://www.cerias.purdue.edu/about/history/coast_resources/idcontent/detection.html.

60). Coates, M., A. O. Hero, et al. (2002). "Internet tomography", IEEE Signal Processing Magazine: 47-65.

61). Cohen, F. (1987). "Computer viruses: Theory and experiments", Computers and Security **6**: 22-35.

62). Cramer, M. L. (1995). "New methods of intrusion detection using control-loop measurement", Technology in Information Security Conference (TISC).

63). Crosbie, M. and E. Spafford (1995). "Applying genetic programming to intrusion detection", Proceedings of the AAAI Fall Symposium.

64). D'Apice, C., Y. Khokhlov, et al. (2010). "Modeling of network traffic", International Conference on the Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), IEEE.

65). Debar, H., M. Becker, et al. (1992). "A neural network component for an intrusion detection system", Proceedings 1992 IEEE Computer Society Symposium on Research in Security and Privacy, May 4-6 1992, Oakland, CA, USA, Publ by IEEE, Piscataway, NJ, USA.

66). Debar, H., D. Curry, et al. (2007). "The Intrusion Detection Message Exchange Format (IDMEF)", Internet Engineering Task Force.

67). Debar, H., M. Dacier, et al. (1999). "Towards a taxonomy of intrusion-detection systems", Computer Networks-the International Journal of Computer and Telecommunications Networking **31**(8): 805-822.

68). Debar, H., M. Dacier, et al. (2000). "A revised taxonomy for intrusion-detection systems", Annales Des Telecommunications-Annals of Telecommunications **55**(7-8): 361-378.

69). Debar, H. and B. Morin (2002). "Evaluation of the diagnostic capabilities of commercial intrusion detection systems", Recent Advances in Intrusion Detection, Proceedings. Berlin, SPRINGER-VERLAG BERLIN. **2516:** 177-198.

70). Denning, D. E. (1987). "An intrusion-detection model", IEEE Transactions on Software Engineering **SE-13**(2): 222-232.

71). Di, W., D. Ji, et al. (2005). "Intrusion detection based on an improved ART2 neural network", Parallel and Distributed Computing, Applications and Technologies, 2005. PDCAT 2005. Sixth International Conference on.

72). Dickerson, J. E. and J. A. Dickerson (2000). "Fuzzy network profiling for intrusion detection", 19th International Confernce of the North American Fuzzy Information Processing Society-NAFIPS (PEACH FUZZ 2000), Jul 13-Jul 15 2000: 301-306.

73). Dickerson, J. E., J. Juslin, et al. (2001). "Fuzzy intrusion detection", Joint 9th IFSA World Congress and 20th NAFIPS International Conference, Jul 25-28 2001, Vancouver, BC, Canada, Institute of Electrical and Electronics Engineers Inc.

74). Duda, R. O., P. E. Hart, et al. (2001). "Pattern Classification", John Wiley & Sons, INC.

75). Elshoush, H. T. and I. M. Osman (2011). "Alert correlation in collaborative intelligent intrusion detection systems - A survey ", Applied Soft Computing **11**: 4349-4365.

76). Esmaili, M., R. Safavi-Naini, et al. (1996). "Evidential reasoning in network intrusion detection systems", Proceedings of the 1996 1st Australasian Conference on Information Security and Privacy, ACISP'96, Jun 24-26 1996, Wollongong, Aust.

77). Estevez-Tapiador, J. M., P. Garcia-Teodoro, et al. (2003). "Stochastic protocol modeling for anomaly based network intrusion detection", Information Assurance, 2003. IWIAS 2003. Proceedings. First IEEE International Workshop on.

78). Estevez-Tapiador, J. M., P. Garcia-Teodoro, et al. (2004). "Measuring normality in HTTP traffic for anomaly-based intrusion detection", Computer Networks **45**(2): 175-193.

79). Fawcett, T. (2003). "ROC graphs - Notes and practical considerations for data mining Researchers". Palo Alto, CA, HP Laboratories.

80). Fayyad, U., G. Piatetsky-Shapiro, et al. (1996). "The KDD process of extractioning useful knowledge from volumes of data", Communications of the ACM **39**(11): 27-34.

81). Feher, C., Y. Elovici, et al. (2012). "User identity verification via mouse dynamics", Information Sciences.

82). Feng, X., D. Wang, et al. (2010). "Analyzing and Correlating Security Events Using State Machine", Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on.

83). Florez, G., S. M. Bridges, et al. (2002). "An improved algorithm for fuzzy data mining for intrusion detection", 2002 Annual Meeting of the North American Fuzzy Information Processing Society Proceedings. New York, I E E E**:** 457-462.

84). Fogla, P. and W. Lee (2006). "Evading Network Anomaly Detection Systems: Formal Reasoning and Practical Techniques".

85). Fox, K. L., R. R. Henning, et al. (1990). "A neural network approach towards intrusion detection". PO Box 98000, Melbourne FL 32902, Harris Corporation, Government Information Systems Division.

86). Fugate, M. and J. R. Gattiker (2003). "Computer intrusion detection with classification and anomaly detection, using SVMs", International Journal of Pattern Recognition and Artificial Intelligence **17**(3): 441-458.

87). Gabriel, R., T. Hoppe, et al. (2009). "Analyzing Malware Log Data to Support Security Information and Event Management: Some Research Results", Advances in Databases, Knowledge, and Data Applications, 2009. DBKDA '09. First International Conference on.

88). Gaikwad, D., S. Jagtap, et al. (2012). "Anomaly Based Intrusion Detection System Using Artificial Neural Network and Fuzzy Clustering", International Journal of Engineering **1**(9).

89). Gao, B., H. Y. Ma, et al. (2002). "HMMs (Hidden Markov models) based on anomaly intrusion detection method", 2002 International Conference on Machine Learning and Cybernetics, Vols 1-4, Proceedings. New York, I E E E**:** 381-385.

90). Gao, F., J. Sun, et al. (2003). "The prediction role of hidden Markov model in intrusion detection", Electrical and Computer Engineering, 2003. IEEE CCECE 2003. Canadian Conference on.

91). Gao, M. and M. Zhou (2003). "Fuzzy intrusion detection based on fuzzy reasoning Petri Nets", System Security and Assurance, Oct 5-8 2003, Washington, DC, United States, Institute of Electrical and Electronics Engineers Inc.

92). Ghadiri, A. and N. Ghadiri (2011). "An Adaptive Hybrid Architecture for Intrusion Detection Based on Fuzzy Clustering and RBF Neural Networks", Communication Networks and Services Research Conference (CNSR), 2011 Ninth Annual.

93). Giura, P. and W. Wei (2012). "Using large scale distributed computing to unveil advanced persistent threats", Science Journal **1**(3): 93-105.

94). Gomez, J. and D. Dasgupta (2002). "Evolving fuzzy classifiers for intrusion detection", Proceedings of the IEEE **2002**.

95). Gong, R. H., M. Zulkernine, et al. (2005). "A software implementation of a genetic algorithm based approach to network intrusion detection", Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2005 and First ACIS International Workshop on Self-Assembling Wireless Networks. SNPD/SAWN 2005. Sixth International Conference on.

96). Gowadia, V., C. Farkas, et al. (2005). "PAID: A Probabilistic Agent-Based Intrusion Detection system", Computers & Security **24**(7): 529-545.

97). Granadillo, G. G., Y. B. Mustapha, et al. (2012). "An ontology-based model for SIEM environments", Global Security, Safety and Sustainability & e-Democracy, Springer**:** 148-155.

98). Group, W. N. R. (2013). "WITS:Waikato Internet Traffic Storage", from www.wand.net.nz/wits.

99). Handley, M., V. Paxson, et al. (2001). "Network intrusion detection: Evasion, traffic normalization, and end- to-end protocol semantics", Usenix Association Proceedings of the 10th Usenix Security Symposium. Berkeley, USENIX ASSOC**:** 115-131.

100). Hansman, S. and R. Hunt (2005). "A taxonomy of network and computer attacks", Computers & Security **24**: 31-43.

101). He, D. and H. Leung (2004). "A novel CFAR intrusion detection method using chaotic stochastic Resonance".

102). Helmer, G., J. Wong, et al. (2001). "A software fault tree approach to requirements analysis of an intrusion detection system", Proceedings Symposium on Requirements Engineering for Information Security, Purdue University.

103). Hertel, C. R. (2004). "Implementing CIFS: The Common Internet File System", Prentice Hall.

104). Hijazi, A. and N. Nasser (2005). "Using mobile agents for intrusion detection in wireless ad hoc networks", Wireless and Optical Communications Networks, 2005. WOCN 2005. Second IFIP International Conference on.

105). Hochberg, J., K. Jackson, et al. (1993). "NADIR: An automated system for detecting network intrusion and misuse", Computers & Security **12**(3): 235-248.

106). Hofmeyr, S. A., S. Forrest, et al. (1998). "Intrusion detection using sequences of system calls", Journal of Computer Security **6**(3): 151-180.

107). Horeis, T. (2003). "Intrusion detection with neural networks - Combination of self-organizing maps and radial basis function networks for human expert integration". IEEE-CIS Student Research Grant Final Report.

108). Hu, P. Z. and M. I. Heywood (2003). "Predicting intrusions with local linear models", Proceedings of the International Joint Conference on Neural Networks 2003, Vols 1-4. New York, I E E E**:** 1780-1785.

109). Hu, W., Y. Liao, et al. (2003). "Robust support vector machines for anomaly detection in computer security", Proceedings of the 2003 International Conference on Machine Learning and Applications, Los Angeles, CA.

110). IETF (1981). "RFC: 793 Transmission Control Protocol".

111). IETF (1992). "RFC: 1305 Network Time Protocol (Version 3) Specification, Implementation and Analysis".

112). Jahanbani, A. and H. Karimi (2012). "A new Approach for Detecting Intrusions Based on the PCA Neural Networks".

113). Jansen, W. A., P. Mell, et al. (1999). "Applying mobile agents to intrusion detection and response", NIST Interim Report (IR) 6416.

114). Jin, H., J. Sun, et al. (2004). "A fuzzy data mining based intrusion detection model", Proceedings of the 10th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS'04).

115). Jing, L., G. Lize, et al. (2012). "A correlation analysis method of network security events based on rough set theory", Network Infrastructure and Digital Content (IC-NIDC), 2012 3rd IEEE International Conference on.

116). Johnson, J. (1958). "Analysis of image forming systems", Proceedings of the Image Intensifier Symposium, US Army Engineering Research Development Laboratories, Fort Belvoir, USA.

117). Julisch, K. and M. Dacier (2002). "Mining intrusion detection alarms for actionable knowledge", 8th ACM International Conference on Knowledge Discovery and Data Mining.

118). Jun, Z., Y. Jiahai, et al. (2005). "Traffic measurement and analysis of TUNET", Cyberworlds, 2005. International Conference on.

119). Kakuru, S. (2011). "Behavior based network traffic analysis tool", Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on, IEEE.

120). Kaur, N. (2013). "Survey paper on Data Mining techniques of Intrusion Detection", International Journal of Science, Engineering and Technology Research **2**(4).

121). Kendall, K. (1999). "A database of computer attacks for the evaluation of intrusion detection systems", Thesis S.B. and M.Eng. --Massachusetts Institute of Technology Dept. of Electrical Engineering and Computer Science 1999.

122). kent, K. and M. souppaya (2006). "Guide to Computer Security Log Management", National Institute of Standards and Technology (NIST).

123). Kim, D. S., H.-N. Nguyen, et al. (2005). "Genetic algorithm to improve SVM based network intrusion detection system", Advanced Information Networking and Applications, 2005. AINA 2005. 19th International Conference on.

124). Kim, G. H. and E. H. Spafford (1994). "The design and implementation of tripwire: A file system integrity checker", ACM.

125). Kim, H., S. Cho, et al. (2004). "Use of support vector machine (svm) in detecting anomalous web usage patterns".

126). Kim, H. S. and S. D. Cha (2004). "Empirical evaluation of SVM-based masquerade detection using Unix commands", Computers & Security.

127). Kirubarajan, T. and Y. Bar-Shalom (2004). "Probabilistic data association techniques for target tracking in clutter", Proceedings of the IEEE **92**(3): 536-557.

128). Kodialam, M. and T. V. Lakshman (2003). "Detecting network intrusions via sampling: A game theoretic approach", IEEE Infocom 2003.

129). Kolias, K., G. Kambourakis, et al. (2011). "Swarm intelligence in intrusion detection: A survey", Computers & Security.

130). Kotenko, I. and A. Chechulin (2012). "Common Framework for Attack Modeling and Security Evaluation in SIEM Systems", Green Computing and Communications (GreenCom), 2012 IEEE International Conference on.

131). Kotenko, I., O. Polubelova, et al. (2012). "The Ontological Approach for SIEM Data Repository Implementation", Green Computing and Communications (GreenCom), 2012 IEEE International Conference on.

132). Kvarnstrom, H. (1999). "A survey of commercial tools for intrusion detection". Goteborg, Sweden, Chalmers University of Technology.

133). Leckie, C. and R. Kotagiri (2002). "A probabilistic approach to detecting network scans", 2002 IEEE/IFIP Network Operations and Management Symposium, Apr 15-19 2002, Florence, Italy, Institute of Electrical and Electronics Engineers Inc.

134). Lee, C. B., C. Roedel, et al. (2003). "Detection and characterisation of port scan attacks".

135). Lee, W. and S. Stolfo (1998). "Data mining approaches for intrusion detection", Proceedings of the 1998 Usenix Security Symposium.

136). Lei, J. Z. and A. Ghorbani (2004). "Network intrusion detection using an improved competitive learning neural network", Communication Networks and Services Research, 2004. Proceedings. Second Annual Conference on.

137). Lei, L. and Z. Ke-nan (2011). "A New Intrusion Detection System Based on Rough Set Theory and Fuzzy Support Vector Machine", Intelligent Systems and Applications (ISA), 2011 3rd International Workshop on.

138). Lei, X. and P. Zhou (2012). "An intrusion detection model based on GS-SVM Classifier", Information Technology Journal **11**: 794-798.

139). Li, Z., A. Das, et al. (2005). "Theoretical basis for intrusion detection", Systems, Man and Cybernetics (SMC) Information Assurance Workshop, 2005. Proceedings from the Sixth Annual IEEE.

140). Liao, H.-J., R. C.-H. Lin, et al. (2013). "Intrusion detection system: A comprehensive review", Journal of Network and Computer Applications **36**: 16-24.

141). Lin, Y.-T., S.-S. Tseng, et al. (2001). "An intrusion detection model based upon Intrusion Detection Markup Language (IDML)", Journal of Information Science and Engineering **17**: 899-919.

142). Lindqvist, U. (2001). "The inquisitive sensor: A tactical tool for system survivability", Supplement of the 2001 International Conference on Dependable Systems and Networks, Goteborg, Sweden.

143). Lindqvist, U. and P. A. Porras (1999). "Detecting computer and network misuse through the production-based expert system toolset (P-BEST)", Proceedings of the 1999 IEEE Symposium on Security and Privacy, May 9-May 12 1999: 146-161.

144). Lippmann, R. (2008). "DARPA 1999 Query", Personal Communication.

145). Lippmann, R., J. W. Haines, et al. (2000a). "The 1999 DARPA off-line intrusion detection evaluation", Computer Networks-the International Journal of Computer and Telecommunications Networking **34**(4): 579-595.

146). Lippmann, R., J. W. Haines, et al. (2000b). "Analysis and results of the 1999 DARPA off-line intrusion detection evaluation", Recent Advances in Intrusion Detection, Proceedings. Berlin, SPRINGER-VERLAG BERLIN. **1907:** 162-182.

147). Lippmann, R., J. Riordan, et al. (2012). "Continuous Security Metrics for Prevalent Network Threats: Introduction and First Four Metrics". Lincoln Laboratory, MIT.

148). Lippmann, R. P. and R. K. Cunningham (2000). "Improving intrusion detection performance using keyword selection and neural networks", Computer Networks-the International Journal of Computer and Telecommunications Networking **34**(4): 597-603.

149). Lippmann, R. P., D. J. Fried, et al. (2000). "Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation", DARPA Information Survivability Conference and Exposition, 2000. DISCEX '00. Proceedings.

150). Liu, Z., G. Florez, et al. (2002). "A comparison of input representations in neural networks: A case study in intrusion detection", 2002 International Joint Conference on Neural Networks (IJCNN '02), May 12-17 2002, Honolulu, HI, Institute of Electrical and Electronics Engineers Inc.

151). Lu, T., K. Zheng, et al. (2012). "A Danger Theory Based Mobile Virus Detection Model and Its Application in Inhibiting Virus", Journal of Networks **7**(8): 1227-1232.

152). Lu, W. and I. Traore (2004). "Detecting new forms of network intrusion using genetic programming", Computational Intelligence **20**(3): 475-494.

153). Lui, C.-L., T.-C. Fu, et al. (2005). "Agent-based network intrusion detection system using data mining approaches", Information Technology and Applications, 2005. ICITA 2005. Third International Conference on.

154). Lukatsky, A. (2002). "Protect your information with intrusion detection", A-List.

155). Lunt, T. F. (1990). "IDES: An intelligent system for detecting intruders", Proceedings of the Symposium: Computer Security, Threat and Countermeasures, Rome.

156). Ma, J. and S. Perkins (2003). "Time-series novelty detection using one-class support vector machines", International Joint Conference on Neural Networks 2003, Jul 20-24 2003, Portland, OR, United States, Institute of Electrical and Electronics Engineers Inc.

157). Mahmood, Z., C. Agrawal, et al. (2012). "Intrusion Detection in Cloud Computing environment using Neural Network", International Journal of Research in Computer Engineering & Electronics **1**(1): 19-22.

158). Mahoney, M. V. (2012). "Network Anomaly Intrusion Detection Research at Florida Tech", from cs.fit.edu/~mmahoney/dist/.

159). Mahoney, M. V. and P. K. Chan (2003). "An analysis of the 1999 DARPA/Lincoln laboratory evaluation data for network anomaly detection", Recent Advances in Intrusion Detection, Proceedings. Berlin, SPRINGER-VERLAG BERLIN. **2820:** 220-237.

160). Manganaris, S., M. Christensen, et al. (2000). "A data mining analysis of RTID alarms", Computer Networks-the International Journal of Computer and Telecommunications Networking **34**(4): 571-577.

161). Mauro, D. and K. Schmidt (2001). "Essential SNMP", O'Reilly.

162). McHugh, J. (2000). "Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection systems evaluations as performed by Lincoln Laboratory", ACM Transactions on Information and System Security (TISSEC) **3**(4): 262-294.

163). Me, L. (1998). "GASSATA: A genetic algorithm as an alternative tool for security audit trail analysis", First International Workshop on Recent Advances in Intrusion Detection, Louvain-la-Neuve, Belgium.

164). Mill, J. and A. Inoue (2004). "Support vector classifiers and network intrusion detection", Fuzzy Systems, 2004. Proceedings. 2004 IEEE International Conference on.

165). Ming-Yang, S., L. Chun-Yuen, et al. (2011). "Genetic-fuzzy association rules for network intrusion detection systems", Fuzzy Systems (FUZZ), 2011 IEEE International Conference on.

166). MIT Lincoln Laboratory. (2012a). "DARPA Intrusion Detection Evaluation", from http://www.ll.mit.edu/mission/communications/CST/darpa.html.

167). MIT Lincoln Laboratory. (2012b). "Documentation", from http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/docs/index.html.

168). MIT Lincoln Laboratory. (2012c). "Intrusion Detection Attack Database", from http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/docs/attackDB.html#ntinfoscan.

169). Modi, C., D. Patel, et al. (2013). "A survey of intrusion detection techniques in Cloud", Journal of Network and Computer Applications **36**: 42-57.

170). Monticelli, A. and F. F. Wu (1985). "Network observability: Theory", IEEE Transactions on Power Apparatus and Systems **PAS-104**(No. 5): 1042-1048.

171). Moore, A. W. and D. Zuev (2005). "Internet traffic classification using bayesian analysis techniques", Proceedings of the ACM SIGMETRICS, Banff, Canada.

172). Moore, K. D., J. S. Jaffe, et al. (2000). "Development of a new underwater bathymetric laser imaging system: L-Bath", Journal of Atmospheric and Oceanic Technology **17**(8): 1106-1117.

173). Moorthy, M. and S. Sathiyabama (2012). "A study of Intrusion Detection using data mining", Advances in Engineering, Science and Management (ICAESM), 2012 International Conference on.

174). Mukkamala, S., G. Janoski, et al. (2002a). "Intrusion detection using neural networks and support vector machines", 2002 International Joint Conference on Neural Networks (IJCNN '02), May 12-17 2002, Honolulu, HI, Institute of Electrical and Electronics Engineers Inc.

175). Mukkamala, S., G. Janoski, et al. (2002b). "Intrusion detection: Support vector machines and neural networks", International Joint Conference on Neural Networks IJCNN'02.

176). Mukkamala, S. and A. Sung (2002). "Feature ranking and selection for intrusion detection systems using support vector machines", Workshop on Statistical and Machine Learning Techniques in Computer Intrusion Detection.

177). Mukkamala, S. and A. H. Sung (2003a). "Artificial intelligent techniques for intrusion detection", System Security and Assurance, Oct 5-8 2003, Washington, DC, United States, Institute of Electrical and Electronics Engineers Inc.

178). Mukkamala, S. and A. H. Sung (2003b). "A comparative study of techniques for intrusion detection", Tools with Artificial Intelligence, 2003. Proceedings. 15th IEEE International Conference on.

179). Mukkamala, S. and A. H. Sung (2003c). "Detecting denial of service attacks using support vector machines", The IEEE International conference on Fuzzy Systems, May 25-28 2003, St. Louis, MO, United States, Institute of Electrical and Electronics Engineers Inc.

180). Mukkamala, S. and A. H. Sung (2003d). "Feature selection for intrusion detection with neural networks and support vector machines", Transportation Security and Infrastructure Protection - Safety and Human Performance. Washington, TRANSPORTATION RESEARCH BOARD NATL RESEARCH COUNCIL**:** 33-39.

181). Muller, K.-R., S. Mika, et al. (2001). "An introduction to kernel-based learning algorithms", IEEE Transactions on Neural Networks **12**(2).

182). National Vulnerability Database. (2011). "CVE and CCE Statistics Query Page", from web.nvd.nist.gov/view/vuln/statistics.

183). Neuman, P. G. and P. A. Porras (1999). "Experience with EMERALD to date", Proceedings of the Workshop on Intrusion Detection and Network Monitoring, Santa Clara, California.

184). Nguyen, B. V. (2002). "An application of support vector machines to anomaly detection", Ohio University.

185). Nicolette, M. and K. M. kavanagh (2011). "Magic Quadrant for Security Information and Event Management", Gartner.

186). Ning, P., X. S. Wang, et al. (2000). "A query facility for common intrusion detection framework", Proceedings of the 23rd National Information Systems Security Conference, Baltimore, MD.

187). North Atlantic Treaty Organization. Research and Technology Organization. (2002). "Intrusion detection generics and state-of-the-art".

188). Ouedraogo, M., D. Khadraoui, et al. (2012). "Appraisal and reporting of security assuarnce at operational systems level", The Journal of Systems and Software **85**: 193-208.

189). Pan, Z., S. Chen, et al. (2003). "Hybrid neural network and C4.5 for misuse detection".

190). Papadaki, M. (2004). "Classifying and Responding to Network Intrusions", Doctor of Philosophy, University of Plymouth.

191). Papoulis, A. (1991). "Probability, Random Variables, and Stochastic Processes", McGraw Hill.

192). Pastrana, S., A. Orfila, et al. (2011). "A Functional Framework to Evade Network IDS", System Sciences (HICSS), 2011 44th Hawaii International Conference on.

193). Patcha, A. and J.-M. Park (2004). "A game theoretic approach to modeling intrusion detection in mobile ad Hoc Networks", Proceedings of the 2004 IEEE Workshop on Information Assurance and Security, United States Military Academy, West Point NY.

194). Patel, A., M. Taghavi, et al. (2013). "An intrusion detection and prevention system in cloud computing: a systematic review", Journal of Network and Computer Applications **36**: 25-41.

195). Patton, S., W. Yurcik, et al. (2001). "An achilles' heel in signature-based IDS: Squeling false positive in SNORT", 4th International Symposium Recent Advances in Intrusion Detection (RAID), University of California-Davis.

196). PCI Security Standards Council (2010). "Payment Card Industry (PCI) Data Security Standard Version 2.0". Requirements and Security Assessment Procedures.

197). Portokalidis, G. and H. Bos (2007). "SweetBait: Zero-hour worm detection and containment using low- and high-interaction honeypots", Computer Networks **51**(5): 1256-1274.

198). Post, G. and A. Kagan (1998). "The use and effectiveness of anti-virus software", Computers & Security **17**(7): 589-599.

199). PricewaterhouseCoopers (2008). "BERR Information Security Breaches Survey - Technical Report", BERR**:** 36.

200). PricewaterhouseCoopers (2012). "Information security breaches survey - Technical report".

201). Provost, F. and R. Kohavi (1998). "Guest editors' introduction: On applied research in machine learning", Machine Learning **30**(2): 127-132.

202). Prowse, J. W. (2013). "Comparing Intrusion Systems", Personal Communication.

203). Ptacek, T. H. and T. N. Newsham (1998). "Insertion, evasion, and denial of service: Eluding network intrusion detection", Secure Networks Inc.

204). Qin, X., D. Dagon, et al. (2004). "Worm detection using local networks", Proceedings of the Recent Advances of Intrusion Detection, RAID '04.

205). Quang, A. T., Q. L. Zhang, et al. (2002). "Evolving support vector machine parameters", 2002 International Conference on Machine Learning and Cybernetics, Vols 1-4, Proceedings. New York, I E E E**:** 548-551.

206). Quang, A. T., Q. L. Zhang, et al. (2003). "Attack recall control in anomaly detection", 2003 International Conference on Communication Technology, Vol 1 and 2, Proceedings. Beijing, BEIJING UNIV POSTS TELECOMMUNICAT PRESS**:** 382-384.

207). Rafsanjani, M. K., L. Aliahmadipour, et al. (2012). "A hybrid Intrusion Detection by game theory approaches in MANET", Indian Journal of Science and Technology **5**(2): 2123-2131.

208). Razo-Zapata, I. S., C. Mex-Perera, et al. (2012). "Masquerade attacks based on user's profile", Journal of Systems and Software.

209). Roesch, M. (1999). "Snort - Lightweight intrusion detection for networks", Proceedings of USENIX 13th Systems Administration Conference (LISA '99), Berkeley, CA.

210). Rossi, M., Ed. (2010). "Symantec Global Internet Security Threat Report - Trends for 2009", Symantec.

211). Salah, S., G. Macia-Fernandez, et al. (2013). "A model-based survey of alert correlation techniques", Computer Networks **57**: 1289-1317.

212). SANS. (2013). "20 Critical Security Controls Version 4.1", 2013.

213). Sasikumar, R. and D. Manjula (2012). "Intrusion Detection System Designed for Wireless using JADE Mobile Agent Framework", International Journal of Computer Applications **50**(15): 23-27.

214). Schwartz, M. and L. Shaw (1975). "Signal Processing", McGraw Hill.

215). Seleznyov, A., V. Terziyan, et al. (2000). "Temporal-probabilistic network approach for anomaly intrusion detection", 12th Annual Computer Security Incident Handling Conference, Chicago, USA.

216). Shah, B. and B. H. Trivedi (2012). "Artificial Neural Network based Intrusion Detection System: A Survey", International Journal of Computer Applications **39**(6).

217). Shameli Sendi, A., M. Dagenais, et al. (2012). "Real Time Intrusion Prediction based on Optimized Alerts with Hidden Markov Model", Journal of Networks **7**(2): 311-321.

218). Shimamura, M. and K. Kono (2006). "Using Attack Information to Reduce False Positives in Network IDS", Computers and Communications, 2006. ISCC '06. Proceedings. 11th IEEE Symposium on.

219). Shipley, G. (2008). "SIEM tools come up short".

220). Skolnik, M. I. (1980). "Introduction to Radar Systems", McGraw-Hill International Book Company.

221). Sommer, P. (1999). "Intrusion detection systems as evidence", Computer Networks-the International Journal of Computer and Telecommunications Networking **31**(23-24): 2477-2487.

222). sommers, J. and P. Barford (2004). "Self-Configuring Network Traffic Generation". Proceedings of the ACM Internet Measurement Conference.

223). Sommers, J., H. Kim, et al. (2004). "Harpoon: a flow-level traffic generator for router and network tests", ACM SIGMETRICS Performance Evaluation Review, ACM.

224). Song, D., M. I. Heywood, et al. (2003). "A linear genetic programming approach to intrusion detection", Genetic and Evolutionary Computation - Gecco 2003, Pt Ii, Proceedings. Berlin, SPRINGER-VERLAG BERLIN. **2724:** 2325-2336.

225). Song, D., M. I. Heywood, et al. (2005). "Training genetic programming on half a million patterns: An example from anomaly detection", Evolutionary Computation, IEEE Transactions on **9**(3): 225-239.

226). Sourcefire (2012). "SNORT and Commercial Products that use it", Personal Communication.

227). Spangler, R. (2003). "Packet sniffer detection with antisniff", University of Wisconsin, Whitewater. Department of Computer and Network Administration.

228). Staniford-Chen, S., S. Cheung, et al. (1996). "GrIDS-a graph based intrusion detection system for large networks", Baltimore.

229). Stevens, W. R. (1994). "TCP/IP Illustrated: the protocols", Addison-Wesley Professional.

230). Sung, A. H. and S. Mukkamala (2003). "Identifying important features for intrusion detection using support vector machines and neural networks", 2003 Symposium on Applications and the Internet, Proceedings. Los Alamitos, IEEE COMPUTER SOC**:** 209-216.

231). Symantec. (2011). "Symantec Introduces New Security Solution to Counter Advanced Persistent Threats", from http://investor.symantec.com/phoenix.zhtml?c=89422&p=irol-newsArticle&ID=1559236&highlight=.

232). Taleck, G. (2003). "Ambiguity resolution via passive OS fingerprinting". RAID 2003.

233). The Cooperative Association for Internet Data Analysis. (2012). "Home Page", from www.caida.org.

234). Thottan, M. and C. Ji (2003). "Anomaly detection in IP networks", IEEE Transactions on Signal Processing **51**(8): 2191-2204.

235). Tian, J.-f., Y. Fu, et al. (2005). "Intrusion detection combining multiple decision trees by fuzzy logic", Parallel and Distributed Computing, Applications and Technologies, 2005. PDCAT 2005. Sixth International Conference on.

236). Tjhai, G., M. Papadaki, et al. (2008). "Investigating the problem of IDS false alarms: An experimental study using Snort". IFIP SEC 2008 Milan, Italy.

237). Tockhorn, A., P. Danielis, et al. (2011). "A Configurable FPGA-Based Traffic Generator for High-Performance Tests of Packet Processing Systems", ICIMP 2011, The Sixth International Conference on Internet Monitoring and Protection.

238). Topallar, M., M. O. Depren, et al. (2004). "Host-based intrusion detection by monitoring Windows registry accesses", Signal Processing and Communications Applications Conference, 2004. Proceedings of the IEEE 12th.

239). Tran, Q.-A. (2004). "One-class support vector machine for anomaly network traffic detection", 2nd Network Research Workshop of the 18th APAN, Cairns, Australia.

240). Tran, T., I. Aib, et al. (2012). "An evasive attack on SNORT flowbits", Network Operations and Management Symposium (NOMS), 2012 IEEE.

241). Tucker, C. J., S. M. Furnell, et al. (2006). "A new taxonomy for intrusion detection". International Networking Conference INC'6. Plymouth.

242). Tucker, C. J., S. M. Furnell, et al. (2007). "A new taxonomy for comparing intrusion detection systems", Internet Research **17**(1): 88-98.

243). Tung, B. (2000). "The common intrusion specification language: a retrospective", DARPA Information Survivability Conference and Exposition, 2000. DISCEX '00. Proceedings.

244). Urick, R. J. (1967). "Principles of Underwater Sound for Engineers", McGraw Hill.

245). Valeur, F., G. Vigna, et al. (2004). "Comprehensive approach to intrusion detection alert correlation", Dependable and Secure Computing, IEEE Transactions on **1**(3): 146-169.

246). Van Trees, H. L. (2001). "Detection, Estimation, and Modulation Theory", Wiley Interscience.

247). Verwoerd, T. (1999). "Active network security", Department of Computer Science, University of Canterbury, New Zealand.

248). Visscher, B. (2007). "Sguil: The Analyst Console for Network Security Monitoring", Squil 0.8.0. 2013, from squil.sourceforge.net/index.html.

249). Wand Network Research Group. (2012). "Case study: MOAT, NLANR", from http://www.wand.net.nz/pubs/19/html/node25.html.

250). Wireshark Foundation. (2012). "Home Page", 2012, from www.wireshark.org.

251). Wu, D. and F. Wong (1998). "Remote sniffer detection". Berkeley, Computer Science Division, University of California.

252). Xiang, G., X. Dong, et al. (2005). "Correlating alerts with a data mining based approach", e-Technology, e-Commerce and e-Service, 2005. EEE '05. Proceedings. The 2005 IEEE International Conference on.

253). Xiao, S. C., J. P. Li, et al. (2005). "Design and analysis of mobile agent for intrusion detection", Wavelet Analysis and Active Media Technology Vols 1-3**:** 330-335.

254). Xu, D. (2006). "Correlation Analysis of Intrusion Alerts", Doctor of Philosophy, North Carolina State University.

255). Yao, J. T., S. L. Zhao, et al. (2005). "A study on fuzzy intrusion detection", Proceedings of SPIE, Vol. 5812, Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security, Orlando Florida, USA.

256). Ye, N., X. Y. Li, et al. (2001). "Probabilistic techniques for intrusion detection based on computer audit data", IEEE Transactions on Systems Man and Cybernetics Part a-Systems and Humans **31**(4): 266-274.

257). Young, G. and J. Pescatore (2010). "Magic Quadrant for Network Intrusion Prevention Systems", Gartner.

258). Zadeh, L. A. (1988). "Fuzzy logic", Computer **21**(4): 83-93.

259). Zhang, X. and Z. Zhu (2004). "Combining the HMM and the neural network models to recognise intrusions", Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai.

260). Zhang, Z., J. Li, et al. (2001). "HIDE: A hierachical network intrusion detection system using statistical preprocessing and neural network classification", Proceedings of the 2001 IEEE Workshop on Information Assurance and Security, West Point, USA.

261). Zhang, Z. and C. Manikopoulos (2003). "Investigation of neural network classification of computer network attacks", Information Technology: Research and Education, 2003. Proceedings. ITRE2003. International Conference on.

262). Zhou, M. and S.-D. Lang (2003). "A frequency-based approach to intrusion detection", Proceedings of the Workshop on Network Security Threats and Countermeasures.

APPENDIX A

THE DARPA 1999 DATASET

Appendix A. The DARPA 1999 Dataset

A.1. Introduction

The DARPA 1999 dataset is one of the most widely used sources of network and host intrusion data. It has been used by many research teams to evaluate both data processing algorithms and complete intrusion systems. The documentation for the dataset can be downloaded from the Lincoln Laboratory website (MIT Lincoln Laboratory 2012b) and comprehensive descriptions of the data collection methodology have been provided by Lippmann (Lippmann, Haines et al. 2000b; Lippmann, Haines et al. 2000a) and Kendall (Kendall 1999).

As considerable published research exists already on the use of the DARPA 1999 dataset, only aspects relevant to the main body of this thesis will be described here. An outline description has already been provided in section 2.6.3.

For this research a data labelling scheme has been adopted to identify which of the many files present in the dataset have been used. A four digit code identifies the week and day number of the recorded files, thus W3D5 corresponds with a TCPDUMP recording taken on Week 3 Day 5. This code was augmented with two additional identifiers to indicate whether the data was recorded on the inside or outside network segment and whether the data was recorded for training or testing purposes. Finally the data label has a prefix of 1999, to differentiate it from the 1998 and 2000 datasets. Thus a file can be identified by a label such as "1999_W2D1_inside_training", which is a DARPA

1999, Week 2, Day 1 recording on the inside network, captured for training purposes.

A single file label identifies the complete frame recording for a given day, on the specified network segment. When it is not necessary to identify the segment or the purpose of the recording, just the initial four digit code is used.

A.2. Description of the Simulated Network

This research is concerned with network intrusion systems and therefore only the TCPDUMP files recorded on the inside and outside of the simulated network were of relevance. Host intrusion data was not used.

Figure A-1 provides an overview of the simulated network.

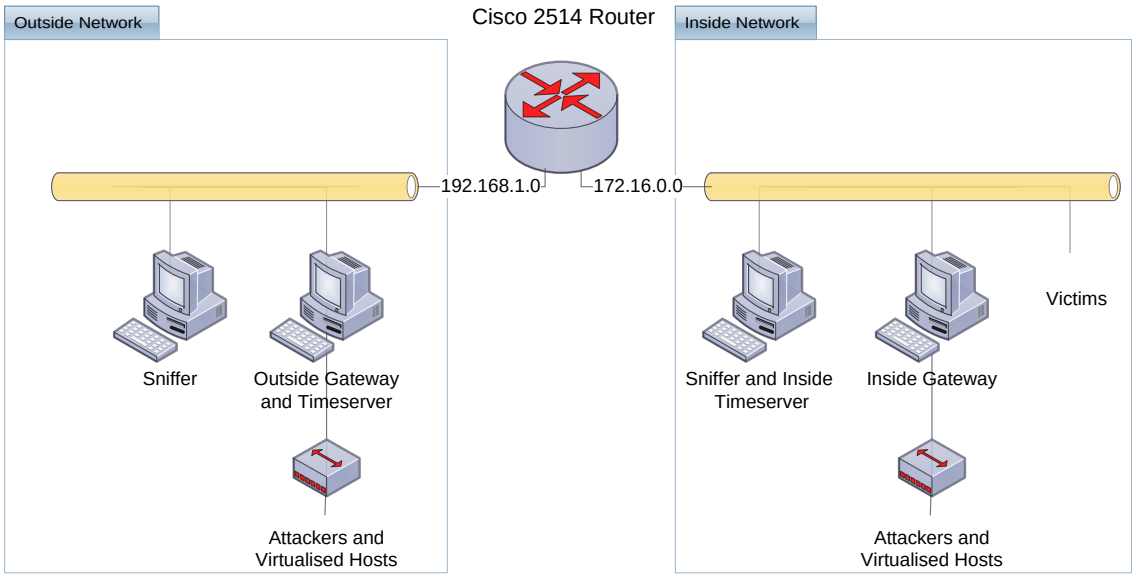


Figure A-1 DARPA 1999 Network, based on (MIT Lincoln Laboratory 2012b)

It can be seen that there are two network segments separated by a router. Both the inside and outside network segments are populated with a variety of operating systems. Separate packet sniffers are present on the two network segments. These sniffers recorded simultaneously the frames present on both

network segments using TCPDUMP. Full frames were recorded including the payload. As the network was simulated the usual privacy concerns did not apply.

A number of attacks were simulated throughout the five weeks of data gathering. These are summarised in Table A-1. It is important to realise that not all of these specific attacks leave a detectable trace in the recorded TCPDUMP files as the experimental setup was also to investigate host based attacks. Further information on each attack can be found at the MIT Lincoln Labs website (MIT Lincoln Laboratory 2012c) and the research of Kendall (Kendall 1999).

Attack Type (No of Attacks)	**OS Specific Attacks**				
	Solaris	**SunOS**	**NT**	**Linux**	**Cisco**
Probe (37)	Queso	Queso	NTinfoscan	Queso	-
	Illegal-sniffer	Illegal-sniffer	Illegal-sniffer	Mscan	-
	IPsweep	IPsweep	IPsweep	LSdomain	-
	Portsweep	Portsweep	Portsweep	Satan	-
DoS (65)	Neptune	Arppoison	Arppoison	Apache2	-
	Pod	Land	Crashiis	Arppoison	-
	Processtable	Mailbomb	Dosnuke	Back	-
	Selfping	Neptune	Smurf	Mailbomb	-
	Smurf	Pod	TCPreset	Neptune	-
	Syslogd	Processtable	-	Pod	-
	TCPreset	-	-	Processtable	-
	Warezclient	-	-	Smurf	-
	-	-	-	TCPreset	-
	-	-	-	Teardrop	-
	-	-	-	UDPstorm	-

Attack Type (No of Attacks)	OS Specific Attacks				
	Solaris	**SunOS**	**NT**	**Linux**	**Cisco**
R2L (56)	Dict	Dict	Dict	Dict	SNMPget
	FTPwrite	Xsnoop	Framespoof	Imap	-
	Guest	-	Netbus	Named	-
	HTTPtunnel	-	Netcat	Ncftp	-
	Xlock	-	Ppmacro	Phf	-
	Xsnoop	-	-	Sendmail	-
	-	-	-	SSHtrojan	-
	-	-	-	Xlock	-
	-	-	-	Xsnoop	-
U2R (37)	Eject	Loadmodule	Casesen	-	-
	FDformat	-	NTFSdos	-	-
	Ffbconfig	-	Makepw	-	-
	Ps	-	Sechole	-	-
	-	-	Xsnoop	-	-
Data (13)	Secret	-	NTFSdos	Secret	-
	-	-	Ppmacro	-	-

Table A-1 Attack Types in the DARPA 1999 Dataset

A.3. Network Statistics

The start and stop dates and times for the TCPDUMP data files are shown in Table A-2. The original documentation provided by the Lincoln Laboratory shows times measured in Eastern Standard Time (EST). Each recorded set of data started early in the morning (typically 8:00am) and finished on the same day, typically 6:00pm. However, EST caused some difficulties in handling the files using GMT or BST as a local time, due to daylight savings time differences. Consequently the information was converted to UTC.

The data in Table A-2 was derived from the outside dataset rather than the inside dataset that was used during this research. This was done as there was

additional data available outside, specifically W3D8. It is important to remember that there is a minor time offset between the inside and outside datasets, as discussed in the next appendix.

Label	Start Date	Start Time (UTC)	End Date	End Time (UTC)	Comments
W1D1	Mar 1	01:00:02 pm	Mar 2	11:00:02 am	Attack free, for training
W1D2	Mar 2	01:00:02 pm	Mar 3	11:00:01 am	Attack free, for training
W1D3	Mar 3	01:00:03 pm	Mar 4	11:00:01 am	Attack free, for training
W1D4	Mar 4	01:00:03 pm	Mar 5	11:00:02 am	Attack free, for training
W1D5	Mar 5	01:00:02 pm	Mar 6	11:00:02 am	Attack free, for training
W2D1	Mar 8	01:00:01 pm	Mar 9	11:00:49 am	Labelled attacks, for training
W2D2	Mar 9	01:00:01 pm	Mar 10	07:59:59 am	Labelled attacks, for training
W2D3	Mar 10	01:00:03 pm	Mar 11	11:00:01 am	Labelled attacks, for training
W2D4	Mar 11	01:00:03 pm	Mar 12	11:00:00 am	Labelled attacks, for training
W2D5	Mar 12	01:00:02 pm	Mar 13	11:00:00 am	Labelled attacks, for training
W3D1	Mar 15	01:00:02 pm	Mar 16	11:00:00 am	Attack free, for training
W3D2	Mar 16	01:00:01 pm	Mar 17	11:00:00 am	Attack free, for training
W3D3	Mar 17	01:00:03 pm	Mar 18	11:00:00 am	Attack free, for training
W3D4	Mar 18	01:00:02 pm	Mar 19	09:11:44 am	Attack free, for training
W3D5	Mar 19	01:00:03 pm	Mar 20	06:02:46 am	Attack free, for training
W3D6	Mar 22	01:00:04 pm	Mar 23	10:14:14 am	Attack free, for training
W3D7	Mar 23	01:00:00 pm	Mar 24	10:59:58 am	Attack free, for training
W3D8	Mar 24	01:00:01 pm	Mar 25	11:00:00 am	Attack free, for training
W4D1	Mar 29	01:00:02 pm	Mar 30	10:59:57 am	Embedded attacks, for testing
W4D2	-	-	-	-	No data is available
W4D3	Mar 31	01:00:09 pm	Apr 1	10:59:57 am	Embedded attacks, for testing
W4D4	Apr 1	01:00:01 pm	Apr 2	10:59:49 am	Embedded attacks, for testing
W4D5	Apr 2	01:00:00 pm	Apr 3	10:59:53 am	Embedded attacks, for testing
W5D1	Apr 5	12:00:02 pm	Apr 6	09:59:56 am	Embedded attacks, for testing
W5D2	Apr 6	12:00:00 Noon	Apr 7	09:59:58 am	Embedded attacks, for testing
W5D3	Apr 7	12:00:00 Noon	Apr 8	09:59:52 am	Embedded attacks, for testing
W5D4	Apr 8	12:00:00 Noon	Apr 9	09:59:53 am	Embedded attacks, for testing

Label	Start Date	Start Time (UTC)	End Date	End Time (UTC)	Comments
W5D5	Apr 9	12:00:04 pm	Apr 10	09:59:58 am	Embedded attacks, for testing

Table A-2 DARPA 1999 Start and Stop Times

Table A-3 provides some basic statistics for each of the inside dataset files. These statistics were produced by a command line utility include with WIRESHARK, known as CAPINFOS.

Label	No Frames (-)	No TCP Conversations (-)	No UDP Conversations (-)	Total No Conversations (-)	Duration (s)
W1D1	1,492,331	39,637	11,363	51,000	79,210
W1D2	1,237,119	46,053	9,559	55,612	79,196
W1D3	1,726,319	46,141	10,339	56,480	79,197
W1D4	1,947,815	41,709	11,590	53,299	79,191
W1D5	1,483,419	37,446	13,304	50,750	79,197
W2D1	1,753,377	44,406	10,974	55,380	79,194
W2D2	1,585,120	56,324	18,699	75,023	68,600
W2D3	1,011,149	26,207	20,364	46,571	79,195
W2D4	1,563,069	68,051	14,092	82,143	79,191
W2D5	1,362,422	49,939	10,639	60,578	79,191
W3D1	2,106,744	43,900	9,374	53,274	79,197
W3D2	1,831,648	50,028	10,246	60,274	79,195
W3D3	1,849,753	48,515	9,666	58,181	79,197
W3D4	1,559,156	19,767	10,979	30,746	72,700
W3D5	1,635,425	61,319	8,092	69,411	61,339
W3D6	1,679,048	12,598	10,931	23,529	76,396
W3D7	2,152,964	49,007	9,850	58,857	79,000
W3D8	No Data	No Data	No Data	No Data	No Data
W4D1	1,647,573	16,790	10,304	27,094	79,195
W4D2	No Data	No Data	No Data	No Data	No Data
W4D3	1,766,074	45,943	14,850	60,793	79,189
W4D4	2,356,503	52,349	14,535	66,884	79,188
W4D5	1,945,538	32,288	17,111	49,399	79,192
W5D1	2,291,319	52,365	9,006	61,371	79,193
W5D2	3,404,824	81,554	10,996	92,550	79,199

Label	No Frames (-)	No TCP Conversations (-)	No UDP Conversations (-)	Total No Conversations (-)	Duration (s)
W5D3	2,087,942	46,198	14,621	60,819	79,192
W5D4	3,201,381	106,690	18,007	124,697	79,192
W5D5	3,393,918	59,581	11,671	71,252	79,193

Table A-3 Statistics for the DARPA 1999 Inside Dataset

The data in Table A-3 was used to calculate the false alarm statistics reported in section 5.4.

APPENDIX B

CLOCK DRIFT IN THE DARPA 1999 DATASET

Appendix B. Clock Drift in the DARPA 1999 Dataset

During this research, initial attempts to match SNORT intrusion alerts with attacks within the TCPDUMP network recording taken from both the inside and outside of the DARPA 1999 simulation network were unsuccessful. It was expected that the truth data provided on the Lincoln Laboratory website (MIT Lincoln Laboratory 2012a) would be definitive and precise, however surprisingly few SNORT alerts matched events in this data. The most likely reasons were that either the matching algorithm was in error, or that SNORT was not detecting the intrusions in the first instance. Testing of the matching software revealed no issues however there did appear to be a time shift between the intrusions that were detected by SNORT and the corresponding truth data entries. This appendix describes the activities undertaken to determine the root cause of the problem and consequently document the magnitude of the effect.

B.1. Initial Analysis

The use of time within the DARPA simulation was examined in detail. An initial attempt to match identical frames in the inside and outside datasets revealed large time differences between the network recordings. The source and destination IP addresses, as well as the absolute TCP sequence numbers were matched for frames near the beginning of each pair of network recordings. As the sequence numbers are unique for a given client or server (IETF 1981; Stevens 1994) matches using these criteria should have indicated corresponding frames precisely. Using this matching technique all the pairs of network recordings for each day of the simulation showed time offsets. Whilst

most of the offsets were only a few seconds, some were much larger with the W2D1 data showing an offset of over 46 seconds.

B.2. Further Analysis

The presence of an offset in time between the simultaneous network recording taken on the inside and outside of the simulated network was surprising and therefore warranted further investigation. WIRESHARK was used to view a limited number of frames in corresponding inside and outside datasets. This showed that there was occasional protocol violations (IETF 1981) when TCP absolute sequence numbers were re-used within a few seconds by a host. Therefore it was necessary to extend the frame matching criteria to include source and destination ports to ensure that false matches did not occur. When this was undertaken the time offsets shown in Figure B-1 were obtained.

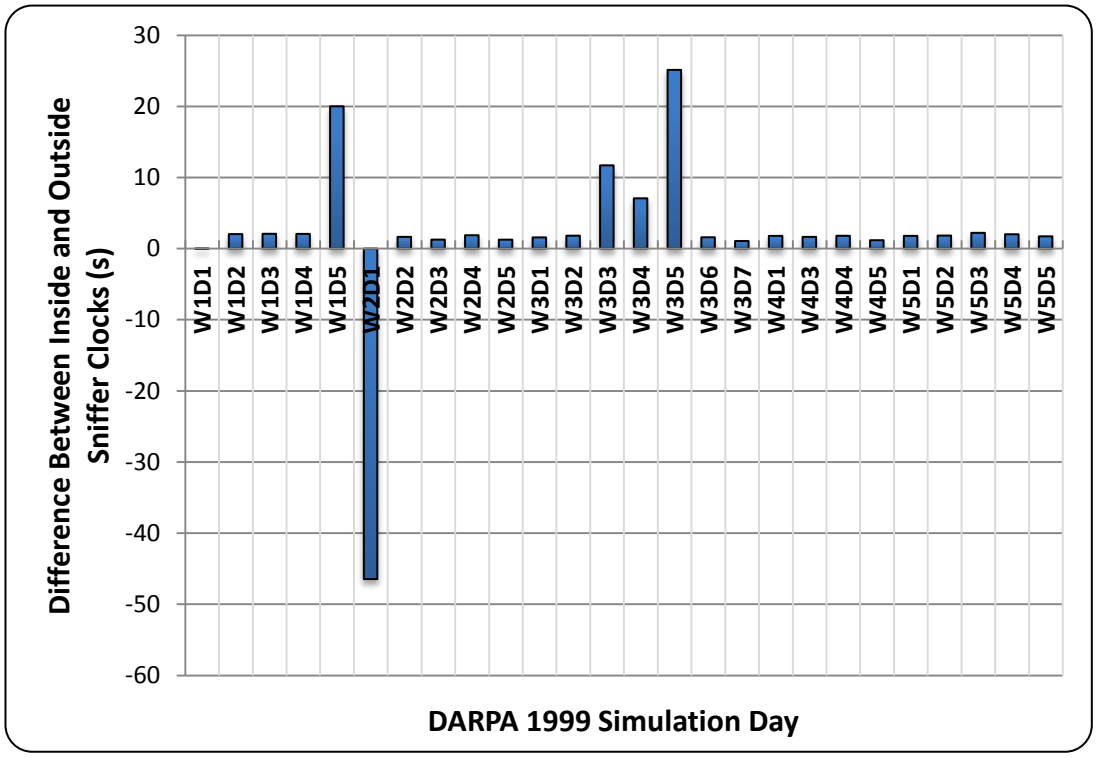


Figure B-1 Clock Drift Between the Inside and Outside Network Sniffers

Further analysis of the network recordings revealed that Network Time Protocol (NTP) (IETF 1992) was configured and running in the DARPA 1999 simulation. Therefore it should be expected that there would be very little or no time offset between the networks. In the outside network, host 192.168.1.10 was configured as the time server and the outside sniffer (192.168.1.90) was in general correctly synchronised to it. No other outside hosts were time synchronised to this NTP server. In the inside network, the NTP server was set as 172.16.112.10, which was also used as the network sniffer. Five other hosts on the inside regularly time synchronised with the inside sniffer. These hosts were the "victim" machines for the simulated network attacks.

The outside network NTP server regularly attempted time synchronisation with the inside NTP server. NTP synchronisation requests were made using "symmetric active" mode, but no corresponding "symmetric passive" frames were sent in response and hence no synchronisation between the inside and outside networks occurred. On the first day of the simulation (W1D1) the inside NTP server also made regular NTP Client Mode requests to the outside NTP server, which responded correctly in NTP Server Mode, allowing time synchronisation to occur between the networks. No further Client Mode requests from the inside NTP server to the outside NTP server are present in the simulation after the first day.

By examining the NTP exchanges between the network time servers the anticipated mode of operation can be determined. This is shown in Figure B-2. It can be seen that the time server and network sniffer are separate physical

servers in the outside network, but combined in the inside network. No evidence of connections to Stratum 0 NTP servers could be located within the frames recorded in either the inside or outside datasets.

The design requirement would have been to synchronise time across all servers and hosts in both the inside and outside networks. This would enable the alignment of events recorded in network sniffer files with host logs, also stored during the simulation. For the current research the most important requirement is the synchronisation between the inside and outside network sniffers, shaded red in Figure B-2.

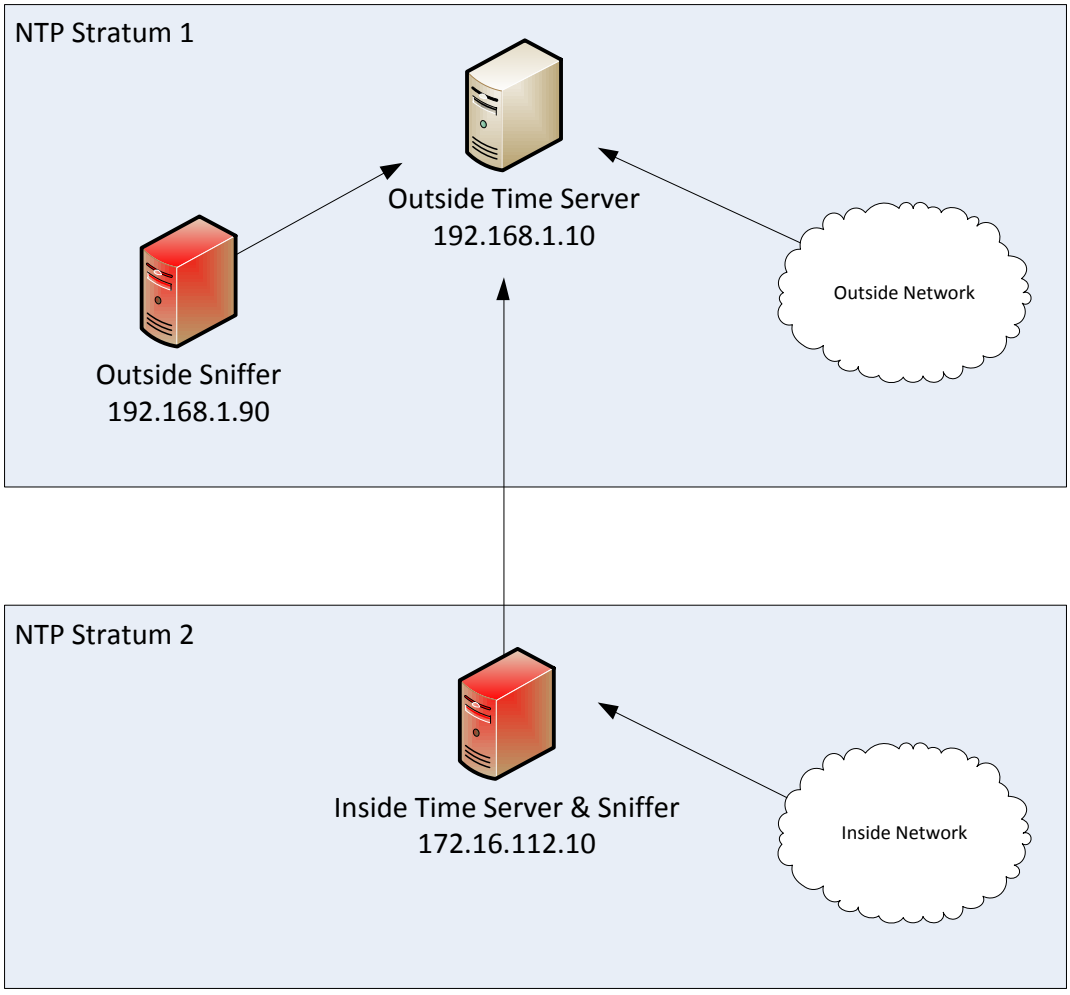


Figure B-2 NTP Hierarchy within the DARPA 1999 Simulation

During the first day of the simulation the network appears to be operating as designed. However, after the first day the inside and outside network sniffers were de-coupled and their clocks drifting with respect to each other. The outside sniffer was synchronised with the outside NTP server, but the inside and outside NTP servers were not synchronised.

In summary, it appears that for the first day of the DARPA 1999 simulation the inside and outside network were correctly synchronised. After the first day, however, they were time synchronised, but not to each other, hence the offset in their times at the start of each days simulation, as highlighted in Figure B-1.

B.3. Clock Drift Measurement

Further work was undertaken to quantify the clock drift between the inside and outside networks during each day of the simulation for which full synchronisation was not operating. TCPDUMP was used to extract specific frames from each of the network files. Only frames from weeks 2, 4 and 5 of the simulation were used as weeks 1 and 3 contained no intrusion events and therefore were not of interest. Connection initiation frames (i.e. with the SYN flag set) were extracted for connections to the principal "victim" hosts, recording the absolute TCP sequence numbers as well as the usual TCPDUMP text output. Microsoft Excel was used to match corresponding frames between the inside and outside network recordings. A single match was taken every hour as representative of the time error and plotted in Figures B-2 to B-4. The initial large timing error for W2D1 persisted for about 36 minutes. It occurred due to a failure of the first seven NTP synchronisations between the outsider sniffer and NTP server. One simulated attack occurred during this period.

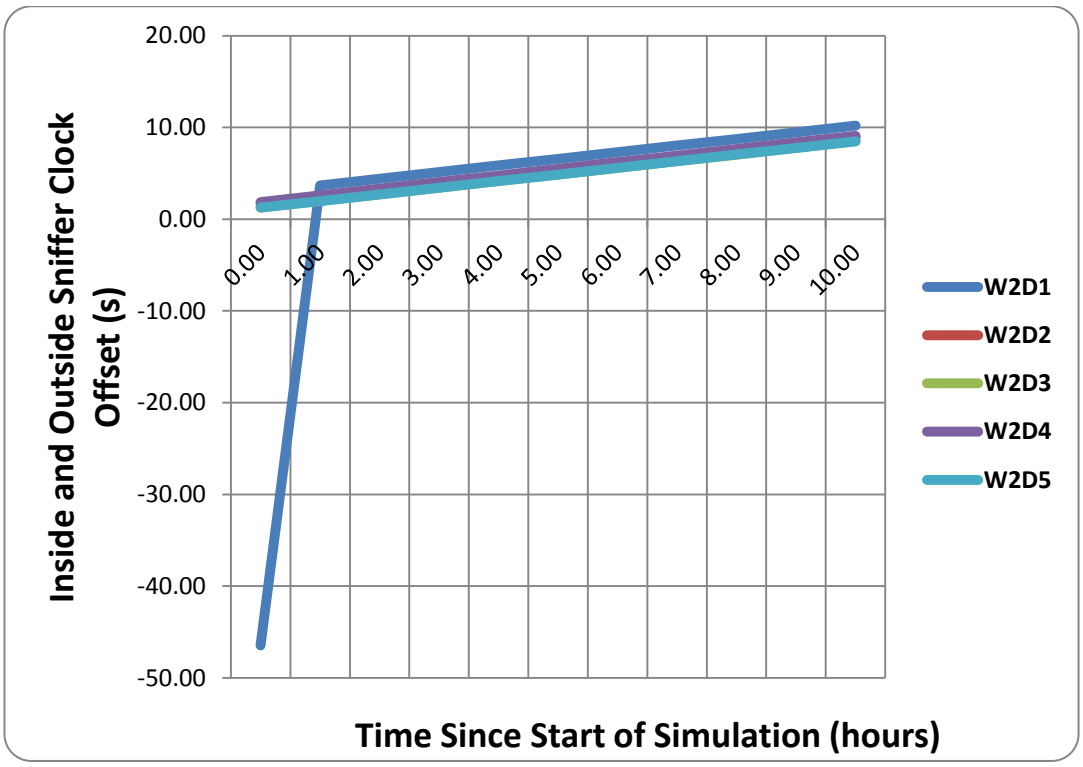


Figure B-3 Clock Drift between the Inside and Outside Network – Week 2

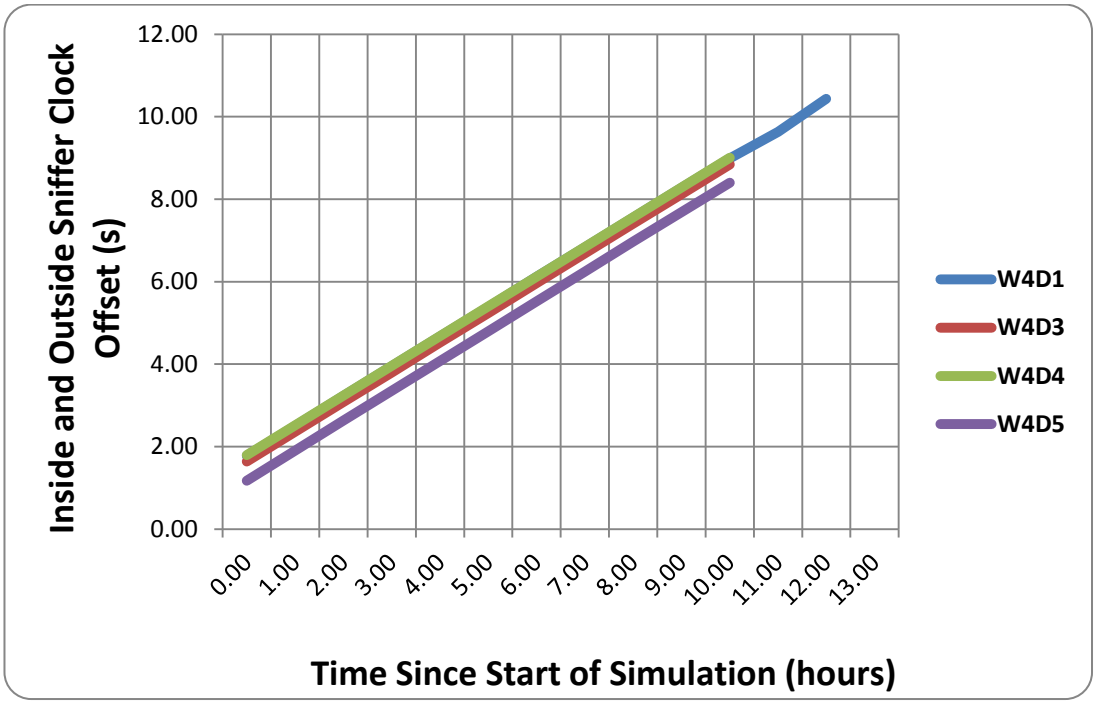


Figure B-4 Clock Drift between the Inside and Outside Network - Week 4

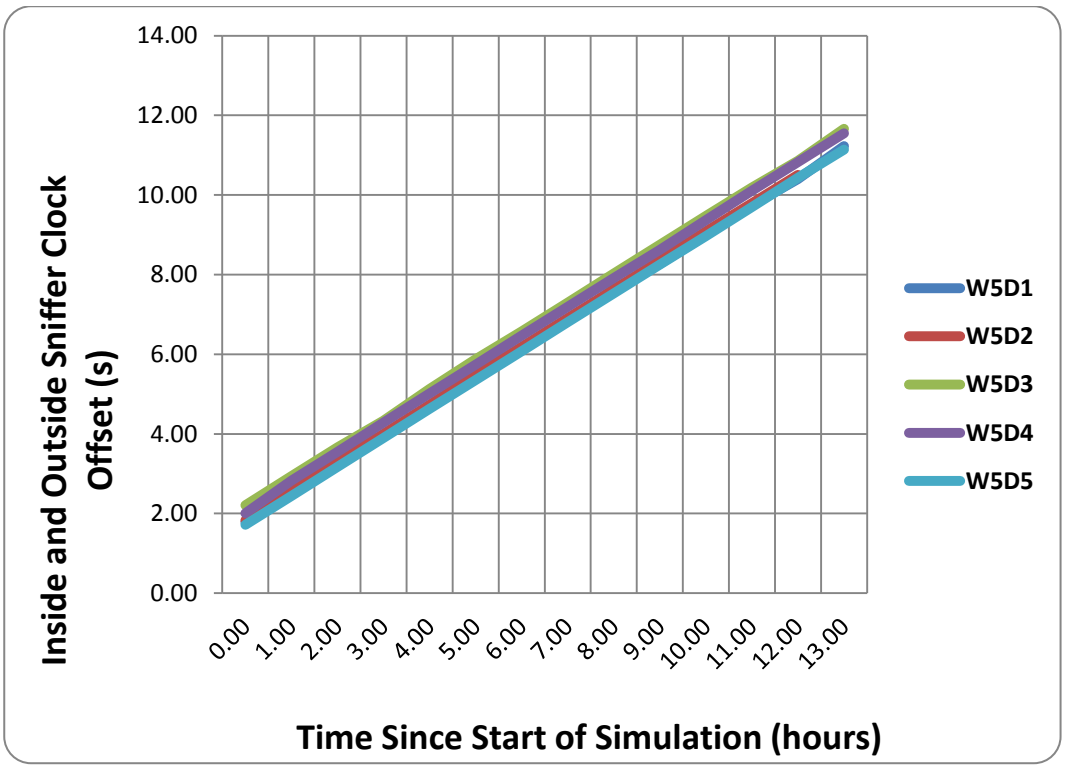


Figure B-5 Clock Drift between the Inside and Outside Network - Week 5

It can be seen that the clock drift is consistent throughout the duration of the simulations. Regression analysis was undertaken with the results shown in Table B-1. This table shows that linear regression appears to be a good representation of the data (i.e. the standard error is low) and the drift in the clocks between the inside and outside networks is approximately 0.73 seconds per hour.

Simulation Day	Slope ($s\ hr^{-1}$)	Intercept ($s\ hr^{-1}$)	Standard Error ($s\ hr^{-1)}$
W2D1	0.7238	2.954	0.00640
W2D2	0.7224	1.634	0.00608
W2D3	0.7239	1.256	0.00628
W2D4	0.7231	1.874	0.00789
W2D5	0.7244	1.249	0.00327
W4D1	0.7211	1.785	0.00432

Simulation Day	Slope ($s\ hr^{-1}$)	Intercept ($s\ hr^{-1}$)	Standard Error ($s\ hr^{-1}$)
W4D2	No Data Available	No Data Available	No Data Available
W4D3	0.7203	1.640	0.00201
W4D4	0.7210	1.794	0.00128
W4D5	0.7229	1.182	0.00625
W5D1	0.7232	1.784	0.03423
W5D2	0.7231	1.844	0.01201
W5D3	0.7253	2.211	0.02636
W5D4	0.7259	2.078	0.02383
W5D5	0.7254	1.721	0.00534

Table B-1 Linear Regression Parameters for Clock Drift

B.4. Conclusions

The lack of time synchronisation between the inside and outside networks in the DARPA 1999 simulation makes matching SNORT alerts with intrusion events in the truth data, across both the inside and outside datasets, problematic. Three obvious approaches to overcome the limitations in the recorded data are:

- Multiple Truth sets, one for each of the inside and outside networks;
- Truth correction, in which the truth data is configured for one of the networks and a correction applied to estimate the truth for the other network. The repeatability of the clock drift as shown in Figures B-3 to B-5 suggests that this approach could be designed to be sufficiently accurate, after the NTP initial synchronisation problems for W2D1 ; and
- Fuzzy time matching, in which precise time is not used as a match criterion. Instead a time window would be used, corresponding to the clock uncertainty.

Although any of these methods could have been made to work it was decided for the research reported in this thesis to limit the analysis to only one of the datasets, eliminating the need to correct for clock drift between the networks.

APPENDIX C

WIRESHARK FOR ATTACK TRUTH DETERMINATION

Appendix C. WIRESHARK for Attack Truth Determination

C.1. Introduction

During this research difficulties were experienced in matching SNORT alerts with specific intrusion events in the DARPA 1999 attack truth data. At first it was thought to be either the result of poor matching software or that the SNORT signatures were not able to respond to the attack types present in the dataset. Further analysis has identified issues with the DARPA 1999 dataset and its truth data. The previous appendix has addressed the clock drift between the inside and outside network segments. This appendix describes the analysis that was undertaken to improve the DARPA 1999 truth data using WIRESHARK measurements.

C.2. Initial Investigation

The initial investigation concentrated on the correctness of the matching software. No issues were identified with the software so effort was directed to the SNORT signatures. Although some attacks produced no alerts from SNORT others produced alerts with minor time discrepancies. Therefore it was decided to examine the recorded frames to see if the attacks could be detected and confirmed manually.

As each days recordings for a given network segment typically contained over one million frames (see Table A-3) a totally manual process was impractical. A frequently used approach to extracting relevant frames from large datasets is to use the filtering functionality of TCPDUMP to pipe frames through additional

processing software. Although this approach could have worked it lacked the ability to visualise attacks, without writing new software. Instead WIRESHARK was used to analysis the datasets for traces of the synthesised attacks.

WIRESHARK has a number of advantages for this task including:

- It is able to load the TCPDUMP files from the DARPA dataset without the need for pre-processing or file conversion software;
- The colour-coded visual display of individual frames was useful in identifying attack elements;
- As WIRESHARK processed the complete frame it was easy to examine in detail the characteristics and bit-settings of frames of interest;
- Included within WIRSHARK is an advanced filtering engine compatible with TCPDUMP, that can be used to display only frames of interest; and
- The frames associated with individual connections could be extracted automatically.

These advantages meant that WIRESHARK was particularly suited to extracting the frames relevant to a given attack. However, its use was not without problems. Versions of WIRESHARK used during early analyses were unable to load a complete set of frames for a whole day without memory errors. Although this was inconvenient, command line software provided with WIRESHARK or filter parameters invoked at start-up were able to limit the number of frames loaded so that memory errors did not occur. More recent versions of WIRESHARK do not appear to have this problem.

WIRESHARK was therefore used to load segments of a recording for a given

network segment and attack frames were located using the existing truth data as a starting point. This was done for all the attack types present in the simulation, and the truth data was updated with measurements derived from the located frames.

C.3. NTinfoscan Analysis

There are a large number of different attack types and instances within the DARPA 1999 dataset. As an illustration of the method and the results that can be obtained this section will describe the analysis for just one attack type, namely NTinfoscan.

The DARPA 1999 intrusion detection attacks database (MIT Lincoln Laboratory 2012c) describes the NTinfoscan as:

"*… a NetBIOS based security scanner. It scans the NT victim to obtain share information, the names of all the users, services running, and other information. The results are saved in an html file named .html where victim is the victim's hostname*."

This database also gives the attack signature information as follows:

"*Sniffing reveals that the attack FTPs to the victim as user anonymous with password guestaccnt@compuserve.com and makes numerous HTML GET requests to files in such directories as /cgi-bin and /scripts. Originally, the ntis ftp'd to the victim with the password, ntinfoscan.*
The security audit log can also be used to detect the attack. A login by IUSR via Advapi, followed by the execution of newdsn.exe by SYSTEM indicates a web scan. A login via KsecDD followed by multiple SAM_USER accesses by SYSTEM indicates a netbios scan."

There are four instances of the NTInfoscan attack, one in the training dataset and three in the testing dataset (attack labels 44.08000, 54.110416 and 54.183002). Examining the first instance in detail, this occurred during W2D1. The truth data provided the following information about this specific attack:

- Start time – 08:01:01 EST (13:01:01 UTC) ;
- End time – Not provided;
- Destination IP – hume.eyrie.af.mil (172.16.112.100);
- Ports Used – Not provided; and
- Source IP – Not provided;

Using WIRESHARK the three phases of the attack could be seen easily. The source of the attack was 206.48.44.18 and using this information the frames associated with each phase could be extracted and the data show in Table C-1.

Attack Phase	**Wireshark Filter**	**No of Frames (-)**	**Start Time (UTC)**	**End Time (UTC)**
FTP/Telnet	ip.addr==206.48.44.18 and ip.addr==172.16.112.100 and (tcp.port == 20 or tcp.port == 21 or tcp.port == 23)	42	13:00:58.1	13:02:02.4
HTTP	ip.addr==206.48.44.18 and ip.addr==172.16.112.100 and tcp.port == 80	90	13:01:59.7	13:17:02.0
Netbios	ip.addr==206.48.44.18 and	1250	13:16:59.9	13:17:01.3

	ip.addr==172.16.112.100 and tcp.port==139			

Table C-1 NTInfoscan Data for W2D1 Inside Network

Table C-1 shows approximately three second time difference between the official DARPA 1999 truth data and that recorded using this method. It can also be seen that the attack duration was over 16 minutes. This approach also identifies the three separate phases of the attack, offering the potential for further insights to SNORT operation against this attack type.

The same filters were applied to the outside dataset, revealing the same number of frames. However the times of events were different, as would be expected from the analysis presented in Appendix B. For example the start of the FTP/Telnet attack occurred at 13:00:11.7, that is 46.4 seconds earlier than the inside dataset.

C.4. Conclusions

WIRESHARK can be used to locate the frames associated with attacks in the DARPA 1999 dataset. Exact timing information can be derived, as well as parameters not within the formal truth data, such as the attack source. Within a given attack the parameters of individual phases can also be extracted if applicable.

WIRESHARK was used during this research to update the formal truth data for the DARPA 1999 dataset which was used to derive the results provided in section 5.4.

APPENDIX D

PERFORMANCE IMPROVEMENT

Appendix D. Performance Improvement

The dominant paradigm for network intrusion systems (NIS) is based on the passive sensing of network traffic, in which the NIS does not interact with other devices on the monitored network. Passive NIS monitor the network sessions between devices, and the frames broadcast throughout the network, from which a determination of an intrusion or non-intrusion event can be made on the basis of known intrusion signatures or anomalous behaviour. This appendix examines one specific method of improving this process further by integrating active probing along with passive sensing, to support the correct assignment between intrusion and non-intrusion events.

Active probing deliberately stimulates network devices into providing more information about their state than can be derived from passive techniques alone. In this aggressive detection process, detection is based on the network or node response to specially designed frame sequences, augmenting the passive interpretation of network activity. This appendix will show that aggressive[3] techniques allow the declaration of intrusion events that are difficult or impossible using data derived solely from passive sensing of a network segment. In this context it offers improvements to the sensitivity of a NIS over conventional passive techniques. Improvements to selectivity are also highlighted.

Given the opportunities for performance improvement it will be necessary to

[3] The term aggressive is proposed as an alternative to the more correct term active, as active intrusion detection already has an accepted alternative meaning.

consider aggressive architectures as an essential part of future high performance network intrusion systems.

D.1. Aggressive Detection

The active paradigm, in which signals with specially designed characteristics are transmitted and the properties of the reflected signal are measured, is well known in other disciplines. In sonar, for example (Urick 1967), military systems often operate in a passive listening mode, sensing the environmental noise and attempting to extract potential target signatures. Passive techniques are preferred as targets are not alerted to the presence of a sonar system, which can then be used to gain a tactical advantage. However, many sonar systems also include an active element in which a sound wave is transmitted and the reflected energy is sensed and processed to confirm the presence of a target. This confirmation is inherently easier from its reflected energy than from its passive signature, but the target can be alerted to the presence of the sonar by its transmitted energy.

Active sonar systems generally have higher sensitivity and selectivity compared with their passive counterparts, at the expense of reduced security, through revealing their presence and location. By optimising the combined use of active and passive techniques many of the security disadvantages of active techniques can be reduced or even overcome, providing significant overall advantage compared with the use of passive techniques alone.

The active paradigm is also used in radar systems (Skolnik 1980), where the improved sensitivity over passive-only radars allows long range detection of

targets of interest. Perhaps less well-known is the application of the active paradigm to television systems, where time gating of the signal from the active element, usually a laser, can achieve the rejection of unwanted signals in the camera (Moore, Jaffe et al. 2000), significantly improving system sensitivity.

The motivation for this research is the extension of these principles to network intrusion systems, by combining active and passive network sensing techniques, to improve sensitivity and selectivity compared to NIS using passive-only techniques. This integration of passive and active detection techniques we call aggressive detection, to differentiate it from purely active or passive systems.

NISs are faced with issues similar to sonar or radar systems. They extract potential intrusion signatures from network measurements that are often gathered passively, to hide the presence of the intrusion system. However, it can be difficult to infer the existence of some intrusions from the passive sensing of network traffic alone. For example, it is hard to determine that a network host has unapproved packet sniffing software installed by an authorised user. This type of intrusion occurs without any unusual frames being transferred over the monitored network and post-installation the user could remove the sensed data using local media without transmitting it over the network. In the experimental work reported in the chapter 5 the "Insidesniffer" attack type could not be detected by SNORT.

The potential of active probing techniques in network security has been recognised by others. Verwoerd noted that a number of active network tools were valuable in the examination of network state (Verwoerd 1999). He

identified some active probing techniques that could be used to assist networking staff in interpreting the alerts produced by an intrusion system. Later Lindqvist (Lindqvist 2001) identified the potential for including active techniques within the discrimination part of an intrusion system. He considered the application of active probing techniques to support the EMERALD intrusion detection system (Neuman and Porras 1999), both in the host and network modules. However, neither of these researchers examined the issues associated with the integration of the active and passive parts of an intrusion engine, which is the subject of this appendix.

D.1.1. Aggressive Network Intrusion Systems

A functional model of an aggressive network intrusion system (AgNIS) is shown in Figure D-1 below. The principle of operation is as follows. The Passive Element senses the frames present on the network making its intrusion/non-intrusion determination in much the same way as in the passive NIS shown in Figure 4-1. Instances of intrusion-like frames however, do not trigger the output of an alert but instead data is transferred to the Active Element, via the Control Channel, so that an active network probe can be selected and transmitted onto the network. The Passive Element continues to acquire frames from the network, receiving the response to the active probe as well as further network activity. If this response confirms that an intrusion event is underway, the Passive Element will output an alert through the Management Channel. If the response confirms that an intrusion event is not underway, the Passive Element will maintain a record in its Potential Intrusion store, so that future probes are not requested, unless there is a further change in state. If the

response neither confirms nor rejects the presence of an intrusion event the Passive Element can take a number of actions, depending on selected strategy including:

- Output an alert, to allow network support staff to investigate further;
- Request additional probes from the Active Element; or
- Wait for further confirmatory network measurements before an alert is output.

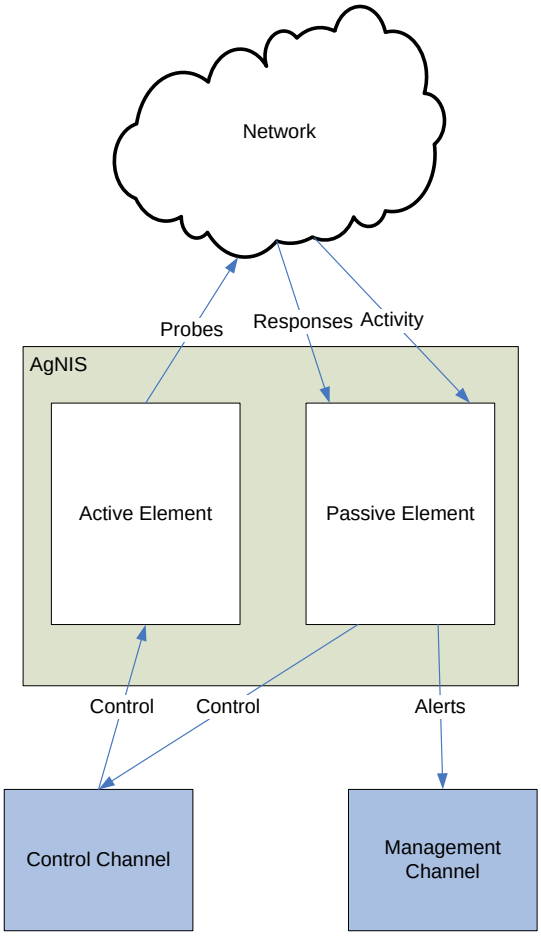


Figure D-1 A Simple AgNIS Functional Description

D.1.2. Active Element Probes

The design and selection of active probes is key to the performance of an AgNIS and should be based the following considerations. First, most networks include management and diagnostic capabilities to assist the support staff deal with network incidents. Such capabilities could provide significant information to a NIS when intrusions are suspected. An obvious approach is for the active element to access SNMP management blocks on suspect devices (Mauro and Schmidt 2001), but an AgNIS could also make information requests to the network management tools directly.

Secondly, the network devices are accessible by the support staff and therefore software could be installed on them, designed to respond to specific, authenticated probes that reveal their security status. This is in many ways similar to the SNMP approach discussed above. However, it offers the potential for more specific information to be passed to the NIS, such as personal firewall logs. This approach will only be effective when the network security policy forbids the connection of personally owned devices, as is frequently the case.

Thirdly, other network devices may be capable of acting as a source of active probes. For example, network vulnerability tools, such as NESSUS, interrogate network devices to confirm their configuration. Vulnerability checks could be monitored by the passive element of an AgNIS to derive additional intrusion information. The infrequent nature of vulnerability checks is unlikely to make this a practical approach and a better way may be to allow the AgNIS to initiate a vulnerability scan on demand. Alternatively, a study of the probes used in vulnerability scanning could reveal useful techniques for direct integration into

the active element of an AgNIS.

Fourthly, probes could be designed to change the behaviour of network devices to improve their data gathering capability during a suspected intrusion event. For example, network sniffers could be turned on and commanded to send data to the passive element when specific frames or sequences of frames occur.

Fifthly, the network tomography techniques developed by Coates (Coates, Hero et al. 2002) may be useful to extract information from traffic measurements made at a small number of network nodes. These techniques are computationally demanding and the probes are network resource intensive, but they may be useful for an AgNIS protecting large-scale networks, such as deployed by Government Agencies or some global enterprises. Further research is necessary to evaluate their potential in the current application; however their ability to extract low-observable measurements is attractive.

Finally, the techniques used by intruders in the reconnaissance stage of an intrusion are designed to reveal detailed information about the target network. Whilst much of this information could be provided a priori to the AgNIS, many of these techniques are sophisticated and potential sources of useful measurements, including:

- Port scanning, to determine a change in state of a network device (Lee, Roedel et al. 2003);
- UDP probes to elicit status information from network devices (Arkin 1999);
- Operating system (OS) finger printing, to confirm that it has not changed

(Taleck 2003);

- MAC / IP address mismatches, to sense a NIC in promiscuous mode (Spangler 2003);
- Decoy services, to tempt intruders into declaring their presence;
- Latency testing, to determine the presence of a denial of service attack, or a NIC in promiscuous mode (Spangler 2003);
- Messages to fictitious hosts, to trigger reverse DNS lookups (Wu and Wong 1998); and
- SMB probes to examine configuration of the host (Hertel 2004).

Clearly there is a wide range of active probes that could be integrated into future network intrusion systems. The next section describes architectures for achieving this integration.

D.2. Architectures for Aggressive Detection

There are many ways that passive and active systems can be integrated to produce an aggressive detection system. In this appendix two characteristics of the integration have been chosen, namely the physical separation and the quantity of information that needs to be transferred between the active and passive elements. The motivation for the selection of these two characteristics stems from the military use of the active paradigm, as discussed earlier. In many military applications the active and passive elements are not co-located. Such systems are generically called bi-static, and offer an improved survivability compared with mono-static or co-located active and passive elements.

Examining the properties of mono-static and bi-static geometries in the current

application provides useful insights, as described later in this appendix, but it is not immediately obvious that this application will deliver all of the usual advantages of bi-static geometries. Specifically the detection of the communications between the active and passive elements could reveal their presence and location to an intruder, if the network that is being protected is used for this communication. As a consequence, the quantity of information was selected as a characteristic for study, as the probability that the passive element will be detected by an intruder will increase with increasing communications between the elements.

When physical separation and quantity of information are combined, four generic architectures result for including active probing within intrusion systems, namely:

- Integrated, Loosely Coupled (ILC) – in which the active element of the intrusion system is combined with the passive intrusion engine, within the same host. The active and passive elements operate independently of each other. However the passive element combines the response of the network to the probes with the passively sensed normal traffic, to make the intrusion/non-intrusion decision. The limited communications between the active and passive elements will not be detectable by an intruder due to their co-location on a single host;
- Integrated, Tightly Coupled (ITC) – in which the active element initiates probes under specific requests from the passive element in response to the detection of potential intrusions. The passive element then combines

the alert data derived from passive-only sensing with the response from the active probes, to amend its intrusion/non-intrusion decision. Both the passive and active elements reside on the same host and therefore the communications between them will not be detectable by an intruder;

- Distributed, Loosely Coupled (DLC) – in which the active element is on a different network node from the passive element of the intrusion engine and operates independently, in the same way as the ILC architecture described above. The separation of the elements will mean that the limited communications between them will be detectable if it occurs over the network that is being monitored; and
- Distributed, Tightly Coupled (DTC) – in which the active and passive elements are on separate nodes within the network but the active probes are sent in response to requests from the passive element. Again, the separation of the elements will mean that the communications between them will be detectable if it occurs over the network that is being monitored.

Each of these approaches has different system-level implications, which will be discussed next.

D.2.1. ILC Architecture

In this architecture the active and passive elements of the intrusion engine are contained on the same host and operate independently of each other. This is illustrated in Figure D-2, where an AgNIS is shown protecting the servers on network segment 1, from unauthorised activity from users on network segment

2.

The passive element senses the frames present on network segment 1. This network segment contains frames from authorised users as well as the probe and response frames from the active element. Whilst the two elements could use the same NIC, they are likely to be present on separate NICs to reduce the vulnerability of the intrusion system. Multiple NICs allow the IP address of the passive element to be disabled, hiding its presence and reducing opportunities for denial of service attacks.

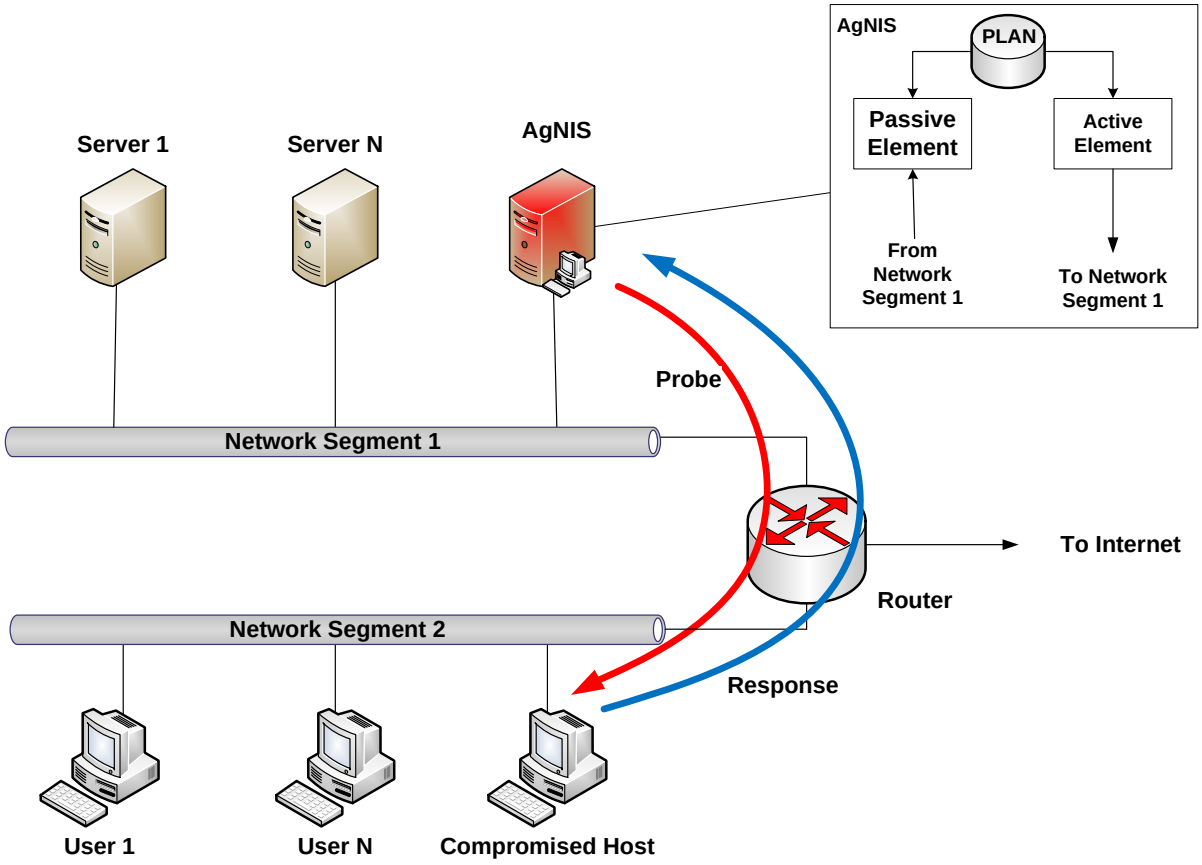


Figure D-2 Integrated, Loosely Coupled AgNIS

The active element sends probes in a pre-set sequence, at selected timings according to a predetermined plan that is shared with the passive element. The

properties of the sequence, such as probe type and frequency, are determined by the support staff in response to their concerns regarding the security of the network. Network devices at higher risk or requiring increased assurance will be probed more frequently and with a wider variety of probes. The selection of the timing interval between probes will be from a consideration of:

- The number of network devices that are being monitored;
- The impact on network bandwidth; and
- The need for randomisation to reduce the vulnerability of the intrusion system to predictive attacks using the probe sequence to penetrate further into the network.

It is not necessary for an intrusion to be detectable by the passive element before active probes are transmitted. This means that ILC systems have the potential to allow detection of low-observable intrusions that are less likely to be detected using passive sensing techniques alone. This is expected to improve the detection rate and as well as reduce the false alarm rate compared with an intrusion system operating solely using passive sensing. The result would be an improvement in sensitivity compared with a passive-only NIS.

The effect of using this architecture on the selectivity of a NIS is unclear at this time. The selectivity associated with low-observable events will have improved, due to their improved detection. However as the probes are not selected to improve discrimination on events detected by the passive element, instead being determined by the pre-agreed plan, it is unlikely that selectivity will generally be improved. More research is required to evaluate this further.

ILC architectures are considered of little interest as they are at significant risk of attack. The active probes alert an attacker to the presence of an intrusion system and identify the host on which it is operating. Penetration or denial of service attacks on the host would be expected. Also, they use network bandwidth for the probes and responses, even during non-intrusion times.

D.2.2.ITC Architecture

In the ITC architecture the active probing element is under the direct control of the passive element, as shown in Figure D-3. The passive element determines potential intrusion events from its sensing of the frames on the network segment 1, which are then passed to the active element for further interrogation.

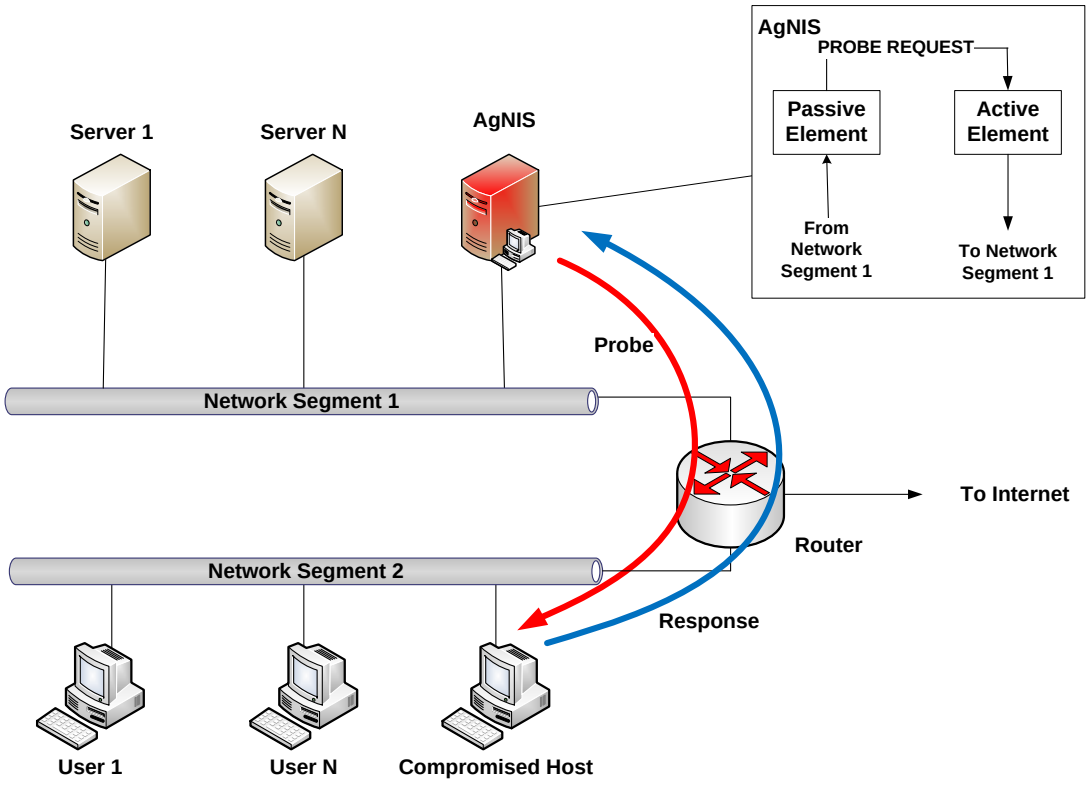


Figure D-3 Integrated, Tightly Coupled AgNIS

The AgNIS combines the results of the passive classifiers with the additional information derived from the active probing to improve the intrusion/non-intrusion discrimination. Again, separate NICs are likely to be needed for the active and passive elements, in order to reduce the vulnerability of the intrusion system.

When operating with this architecture, the active probes provide information to confirm or reject of the presence of an intrusion, initially detected using passive techniques. Therefore it is unable to detect low-observable intrusions directly, but is able to provide the method for a reduction in false alarms.

The sensitivity of the AgNIS can be increased over the use of passive techniques, in the following way. The passive element would operate at a low decision threshold level, equivalent to moving to the right hand side of a ROC curve. This will produce a high detection rate for intrusion events, but also a large false alarm rate. In a NIS using passive techniques alone this false alarm rate is likely to be too high for support staff to investigate effectively. However, the active element of the AgNIS is automated and would therefore be capable of rapid interrogation of each of these passively detected alerts. Only detections confirmed by the response to the active element are output as alerts. In this way the sensitivity of the AgNIS is increased, by moving the operating point on the passive element ROC curve to the right and using two-stage detection to reduce the overall false alarm rate to an acceptable level.

The selectivity of the AgNIS can be increased over the use of passive techniques, by choosing active probes specifically in response to the signatures

triggered in the passive element. For example consider the situation in which a set of signatures have been triggered for a given connection between network devices. The probability vector with unity entries for the triggered signatures could be used calculate the distance to specific intrusion event types in the selectivity matrix. If there is uncertainty about which intrusion event type should be declared, for example by the distances being closer than a threshold value, specific active probes could be dispatched to improve the discrimination. Further probes could also be used if the initial selection failed to provide sufficient discrimination between the possible intrusion event types.

As a specific example of selectivity improvements consider the situation where the passive element detects activity from a specific host that could be interpreted as attempts to force a network switch into hub mode. The active element could be instructed to use probes to determine if the host NIC was in promiscuous mode. If this is confirmed, it is logical to deduce that network sniffing is being attempted and issue a specific alert for action by the network support team. If the host is not in promiscuous mode, further probes may be sent to other hosts on the segment to see if a coordinated attack is underway, involving multiple attackers. Other probes could be considered to provide more selective information on the type of attack underway.

The principal advantage of the ITC architecture is its simplicity. Once the passive part has detected a potential intrusion, the type and parameters of active probes can be selected and initiated against the suspect network device. As the responses from the initial probes are received and interpreted, further

probes can be sent to address any remaining uncertainty. No communication difficulties are present between the active and passive elements due to their co-location on the same host. Thus tight integration of the active and passive parts can be achieved. Additional advantages include:

- Fast response – the tight coupling minimises the communications delays and allows the active element to be responsive to the threat. Alerts are likely to be declared more quickly with a tightly coupled architecture; and
- Resource Efficient - Network resources are used only when the network threat level has increased due to potential intrusions being found by the passive element.

The principal disadvantage of this approach is the vulnerability of the intrusion system to attack. Again, the active probes will identify the host running the intrusion system providing valuable information to an intruder.

D.2.3. DLC Architecture

In the DLC architecture the active and passive elements operate independently on different hosts, as shown in Figure D-4. In this figure the probes are initiated from a host within the same network segment as the suspected compromised host. In general this is not necessary, however the active element must spoof its IP address to direct probe responses to the network segment where the passive element is present.

Many of the features of the ILC architecture apply including:

- Sensitivity is improved;

- The active element would follow a pre-set sequence of probes at set timings, defined by the pre-shared plan;
- The potential to detect low-observable intrusion events is present; and
- There is inefficient use of network resources.

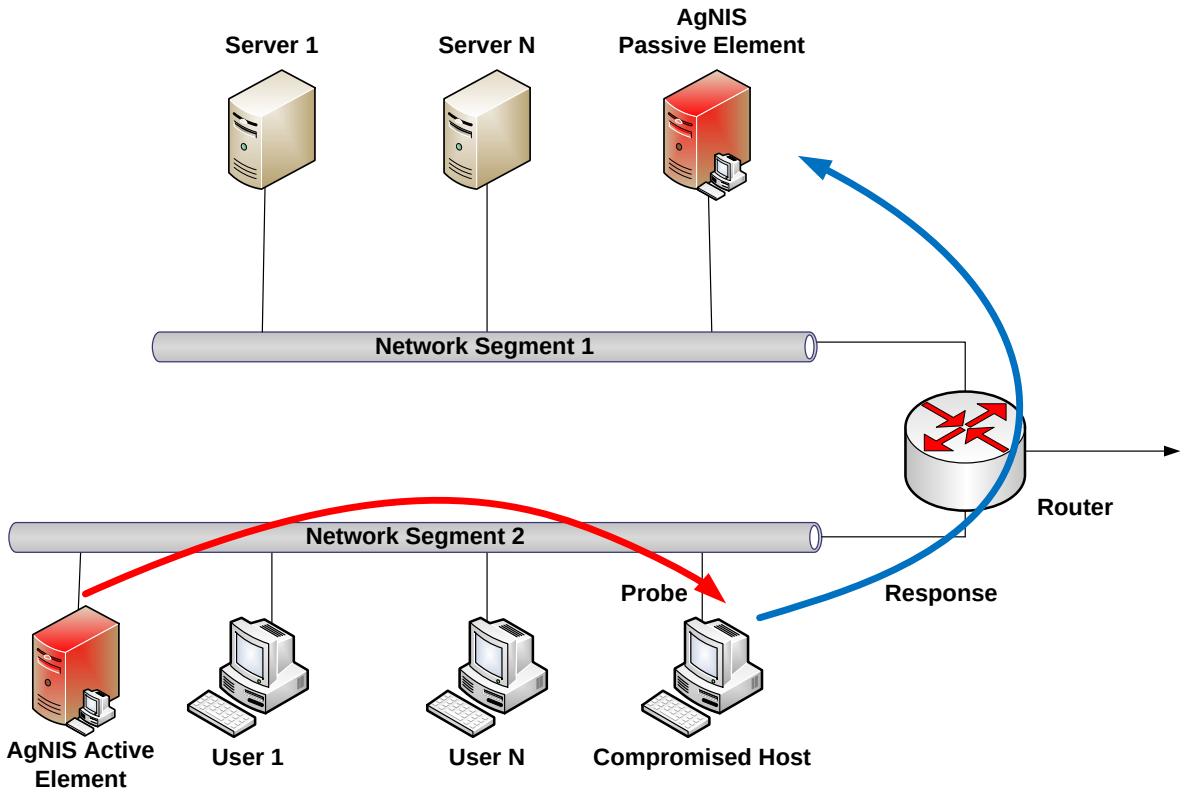


Figure D-4 Distributed, Loosely Coupled AgNIS

There is an important difference however, in terms of the vulnerability of the intrusion system. If the active element is attacked and disabled, the integrity of the intrusion system is not completely lost. The passive element can continue to operate, albeit at reduced performance. The active probes no longer identify the hosts for the intrusion system and therefore attacks against it would be more challenging to initiate.

At first sight the need to spoof the addresses of the active element could reveal the network segment on which the passive element resides. This would be a serious issue, as the passive element provides the executive control of the AgNIS, determining when alerts are issued to support staff. There are a number of ways of overcoming this, including:

- The active element could spoof its addresses to multiple network segments, in sequence, thereby hiding the actual segment on which the passive element resides;
- Honeynet devices could be placed within the network segment being spoofed by the active element. These devices could broadcast alerts when subject to penetration attempts or other NIS defeat techniques, as this would be confirmation of an on-going attack within the network;
- Multiple active and passive elements could be deployed, increasing the survivability of the AgNIS; and
- Active element decoys could be deployed, with honeynet features. These decoys could generate probes with spoofed addresses that do not contain passive elements

The principal disadvantages of the DLC are its inefficient use of network resources, due to its use of network probes when no intruder is present, and reduced responsiveness compared with tightly coupled architectures.

D.2.4. DTC Architecture

In the DTC architecture the active and passive elements are also separated within the network, as shown in Figure D-5. The control of the active element

is determined by messages from the passive element, in much the same way as the ITC architecture. Many of the features of the DLC and ITC architectures are inherited:

- The active system responds only to alerts detected by the passive element, making the principal benefit that of improved sensitivity and selectivity;
- Multiple active elements can be deployed, directing the responses to their probes to the same passive element;
- Reduced vulnerability and increased survivability compared with integrated architectures, due to the redundancy within the AgNIS; and
- Efficient use of network resources, as probes are only initiated in response to a perceived attack.

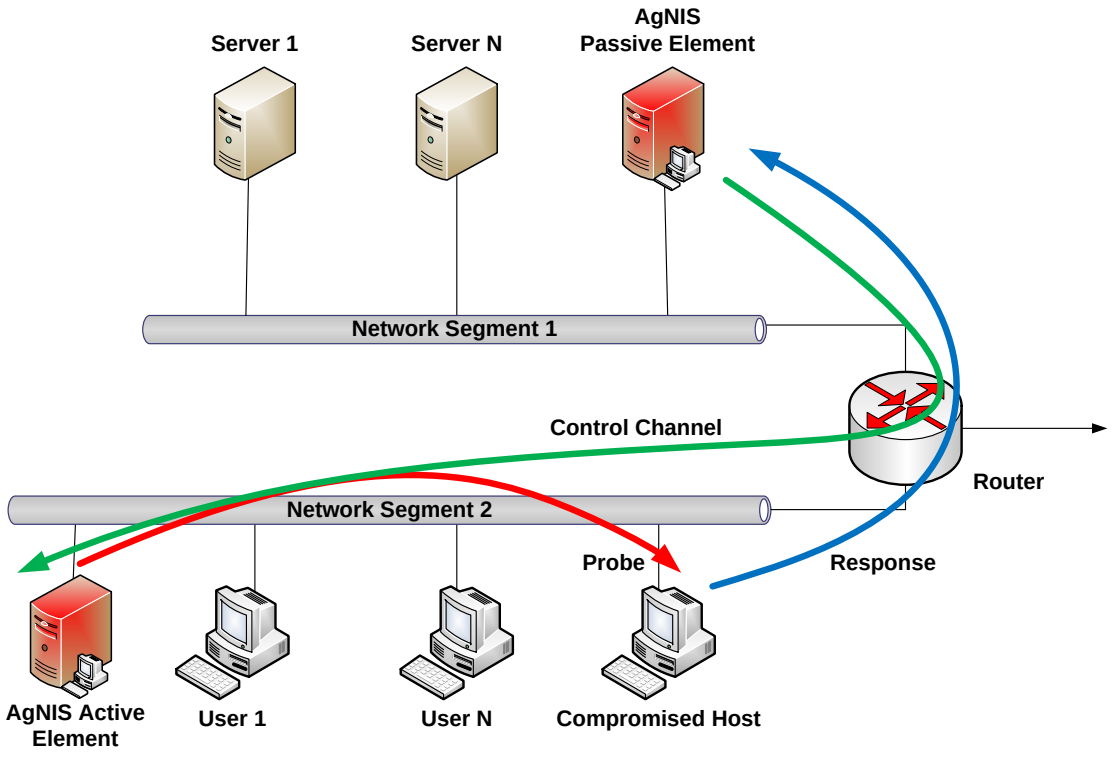


Figure D-5 Distributed, Tightly Coupled AgNIS

The key issue in the DTC architecture is the mechanism for achieving communications. The simplest method is to allow the passive element to communicate with the active element via the monitored network. However, this approach would identify the passive element to a potential attacker and is therefore not considered appropriate. It is common practice for a dedicated VLAN to be used for network management tasks; however such connections are subject to the same security concerns. IPSec tunnels, whilst securing the payload contents, reveal the tunnel end points where the passive and active elements reside.

An alternative communication method could use out-of-band connections, effectively a separate network to that which is being monitored by the AgNIS. Direct peer-to-peer connections between the active and passive elements can be easily achieved. Close physical separation is usually required and likely to be available, even within a large global network, mainly from network administration considerations.

In systems where the out-of-band communication is impractical it would be necessary to consider special techniques to hide the presence of the communication channels on the monitored network. It is proposed to use covert channels to convey the selection of an active probe and its parameters (Ahsan 2002). Such channels are difficult to detect, hiding their presence from intruders sniffing the monitored network. Usually the potential for covert channels to be present is a significant security concern. However, it is believed their use to hide the presence of AgNIS communications to be novel and an

appropriate use of this intruder technique. Covert timing channels are likely to be more applicable than covert storage, providing the channel bandwidth can be sufficiently high to convey the full message without excessive delay.

D.2.5. Hybrid Architectures

Hybrid architectures could also be deployed, in which the active element initiates probes according to a pre-agreed plan controlling the sequence and timing, unless otherwise directed by the passive system. This approach would inherit many of the advantages of the loosely coupled systems as well as maintaining the advantages of the tightly coupled systems. It would be able to detect low-observable intrusion events, during its loosely coupled operation and become highly responsive with a low false alarm rate when operated within its tightly coupled mode. In addition, in systems where multiple active elements have been deployed, some could be tightly coupled whilst others are loosely coupled, achieving the same performance simultaneously.

Hybrid architectures are versatile, offering many options to the network security staff. In loosely coupled mode the active element could be commanded into one of many pre-set plans according to the threat perceived by the passive element. The threat could be determined by the number of alerts being detected by the passive element, the presence of anomalies such as network load or even time of day. Network resource intensive techniques could be deployed in some plans, when the threat was determined to be high or the impact on users could be tolerated. In tightly coupled mode the passive element could command multiple active elements to initiate the probes necessary for discrimination of intrusion packages. This would give redundancy,

improving the survivability of the AgNIS, but it could also be used to obscure the presence of the active elements.

It is likely that hybrid, distributed architectures will offer the most potential for future active network intrusion systems. Their potential for attack resilience as well as their capacity for detecting low-observable intrusions will be important in practical systems.

In Table D-1 below the advantages and disadvantages of the different architectures are summarised and contrasted with a NIS using passive techniques only.

Architecture	Advantages	Disadvantages
Passive Only	• Simple • Minimal hardware requirements • Difficult to detect the deployment of NIS	• Limited sensitivity • Poor false alarm rate • Unable to detect low-observable intrusion events • Poor NIS survivability
Integrated, Loosely Coupled (ILC)	• Simple • Minimal hardware requirements • Detection of low-observable intrusion events • Improved sensitivity	• Vulnerable to direct attack • Inefficient use of network resources • Poor response time (need to wait for discriminating probe) • Poor NIS survivability
Integrated, Tightly Coupled (ILC)	• Simple communications • Improved sensitivity • Improved selectivity • Efficient use of network resources • Fast response	• Detection of low-observable intrusion events is not improved • Host is vulnerable to attack • Poor NIS survivability
Distributed, Loosely Coupled (DLC)	• Detection of low-observable intrusion events • Improved sensitivity	• More hardware required • Inefficient use of network resources

	• Reduced vulnerability to direct attack • Increased NIS survivability • Multiple active elements can be used with a single passive element	• Poor response time (need to wait for discriminating probe)
Distributed, Tightly Coupled (DTC)	• Efficient use of network resources • Improved sensitivity • Improved selectivity • Fast response • Reduced vulnerability to direct attack • Increased NIS survivability • Multiple active elements can be used with a single passive element	• Covert communications required • More hardware required • Detection of low-observable intrusion events is not improved
Hybrid	• Efficient use of network resources • Detection of low-observable events • Fast response • Improved sensitivity • Improved selectivity • Reduced vulnerability to direct attack • Increased NIS survivability • Multiple active elements can be used with a single passive element	• More hardware required • Covert communications required

Table D-1 The Advantages and Disadvantages of AgNIS Architectures

D.3.AgNIS Considerations

There are some important issues associated with active probing that need to be considered before inclusion within deployed intrusion systems. This section describes many of these considerations.

D.3.1. Batch Processing

Lindqvist considered the integration of active probes into EMERALD and he concluded that the intrusion engine must operate in real-time rather than batch mode (Lindqvist 2001). This is true for an intrusion system with tightly coupled active and passive elements, including hybrid approaches. However this is not generally true as architectures can be proposed in which batch processing can also be a potential operating mode. Both the ILC and DLC architectures discussed above can be operated in batch mode. Such architectures may therefore become important in networks where forensic analysis of events is essential.

D.3.2. Network Security and AgNIS Protection

Intrusio systems are generally vulnerable to attack or evasion techniques and it is therefore important that the integration of active probing does not degrade this situation further. Specific techniques will need to be included to protect the NIS against existing and specially developed attacks against AgNIS. For example, in the loosely coupled architectures, the intrusion system could be vulnerable to an attacker injecting a background of false probe responses. This could be addressed by time synchronising the active and passive elements. The passive element would then only include probe responses within a narrow time window. Time synchronisation could be achieved by a covert channel from the active to the passive elements, at the cost of increasing the coupling between them.

Of particular concern is the potential for an attacker to initiate a large number of alerts within the passive part of an AgNIS using tight coupling, using tools

such as STICK (Patton, Yurcik et al. 2001). The resulting number of probes that could be generated would overload the network resources, creating a self-induced denial of service. It is therefore necessary to include features to limit the resources consumed by the active probes. Although this will offer the potential for an attacker to overload the intrusion system with spoofed attacks before initiating an attack on the real target, the presence of an intruder within the network would be obvious. This discussion indicates that care in the selection of security measures and countermeasures is required. It is hoped that this will be subject of future research.

The location of the active elements relative to the passive elements needs careful consideration. It is not necessary for the active element to be on the same subnet as the passive element, nor on the subnet that includes the monitored devices. The active element could spoof its addresses to direct responses at individual passive elements. This is a useful capability and offers the potential for cooperation between multiple passive elements, further improving the survivability of the intrusion system when under attack.

D.3.3. Personal Firewalls

Many organisations deploy personal firewalls on network hosts, either as part of the operating system or as a dedicated security tool. These have the ability to limit the information that can be gathered from network probes and can potentially increase the workload of support staff as they respond to queries raised by network users. At first sight this might appear to limit the applicability of active techniques; however the following approach can be taken:

- Active probes can be used to monitor changes to the open ports and available services on protected hosts, to confirm that such changes are in line with the network security policy;
- Not all the network devices have personal firewalls. Routers, switches, firewalls (low security connections) and servers generally do not. Probes will need to be able to extract the maximum of available information from such devices;
- The personal firewalls are authorised by the network security staff and subject to policy settings. The security team could allow the personal firewalls to respond to specific probes, so as to confirm the status of the host; and
- Many probe types are designed to illicit a response from the IP stack of the host rather than to penetrate it. Application layer firewalls should not affect the responses from lower levels in the stack.

D.3.4. Efficiency

Aggressive NIS have a much improved efficiency compared with passive systems alone. In passive NIS the ratio of frames that are capable of correctly asserting the status of a network or network device may be greater than 1:50,000, as discussed in section 4.5. However, probe responses are designed specifically for this task and by increasing their frequency this ratio can be significantly reduced. The exact reduction depends on the design parameters of the AgNIS, but for a design goal of no more than one active response frame for every 100 normal user frames, to minimise the impact on network bandwidth, this ratio could reduce to 1:100.

The two-stage alerting process, that is inherent in AgNIS architectures that exploit tight coupling, changes the design requirements for the passive element. The goal of the passive element is now to make sure that intrusions are alerted in the first stage, not to reduce the false alarms through the use of sophisticated discrimination techniques. Discrimination is now undertaken in the second stage, where the responses to specific probes provide data that simplify this process. The first stage, passive processing may be achieved with less computationally demanding techniques than in conventional NIS. This further improves efficiency of a NIS as the majority of the NIS measurements are undertaken with simplified processing.

D.4. Summary and Conclusions

Four discrete architectures for integrating active probes with a passive NIS have been investigated and their properties determined. It is clear that an AgNIS approach can offer both sensitivity and selectivity improvements compared with a passive only approach, depending on the implemented architecture. The selection of a particular architecture depends on the detailed requirements including the need for real-time operation, the implications on the network bandwidth and the security concerns of the network owner.

The most attractive approach for real-time operation is a hybrid architecture consisting of multiple distributed active and passive elements communicating is a discreet way. This approach appears to offer the potential to achieve all the benefits of AgNIS architectures whilst minimising their disadvantages.

The security implications of deploying an AgNIS have been reviewed, with the potential for communications between the active and passive elements to reveal their locations identified as a prime concern for distributed systems. Ideally this issue can be resolved with out-of-band connections but when this is not possible the novel approach of using covert channels to achieve discreet communications between the AgNIS elements has been proposed.

The application of AgNIS requires more research to establish how far sensitivity and selectivity can be improved. In particular the discrimination offered by different active probe types needs to be established.

APPENDIX E

SNORT CONFIGURATION

Appendix E. SNORT Configuration

The following SNORT configuration file was used to produce the results reported in Chapter 5. The comments have been removed to reduce the size of this appendix.

```
var HOME_NET [192.168.1.0/24,172.16.0.0/16]
var EXTERNAL_NET any
var DNS_SERVERS $HOME_NET
var SMTP_SERVERS $HOME_NET
var HTTP_SERVERS $HOME_NET
var SQL_SERVERS $HOME_NET
var TELNET_SERVERS $HOME_NET

portvar HTTP_PORTS
[80,2301,3128,7777,7779,8000,8008,8028,8080,8180,8888,9999]
portvar SHELLCODE_PORTS !80
portvar ORACLE_PORTS 1521

var AIM_SERVERS
[64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,64.12.163.0/24,64.12.200.0/24,2
05.188.3.0/24,205.188.5.0/24,205.188.7.0/24,205.188.9.0/24,205.188.153.0/2
4,205.188.179.0/24,205.188.248.0/24]

var RULE_PATH /home/christ/Desktop/PhD/rules
var SO_RULE_PATH /home/christ/Desktop/PhD/rules/so_rules
var PREPROC_RULE_PATH /home/christ/Desktop/PhD/rules/preproc_rules

config disable_decode_alerts
config disable_tcpopt_experimental_alerts
config disable_tcpopt_obsolete_alerts
config disable_tcpopt_ttcp_alerts
config disable_tcpopt_alerts
config disable_ipopt_alerts
config checksum_mode: all

config pcre_match_limit: 1500
config pcre_match_limit_recursion: 1500
config detection: search-method ac-bnfa max_queue_events 5
config event_queue: max_queue 8 log 3 order_events content_length

dynamicpreprocessor directory /usr/local/lib/snort_dynamicpreprocessor/
dynamicengine /usr/local/lib/snort_dynamicengine/libsf_engine.so
dynamicdetection directory /usr/local/lib/snort_dynamicrules
```

```
preprocessor frag3_global: max_frags 65536
preprocessor frag3_engine: policy windows timeout 180

preprocessor stream5_global: max_tcp 8192, track_tcp yes, track_udp no
preprocessor stream5_tcp: policy windows, use_static_footprint_sizes, ports
client 21 22 23 25 42 53 79 80 109 110 111 113 119 135 136 137 139 143 110
111 161 445 513 514 691 1433 1521 2100 2301 3128 3306 6665 6666 6667
6668 6669 7000 8000 8080 8180 8888 32770 32771 32772 32773 32774 32775
32776 32777 32778 32779, ports both 443 465 563 636 989 992 993 994 995
7801 7702 7900 7901 7902 7903 7904 7905 7906 6907 7908 7909 7910 7911
7912 7913 7914 7915 7916 7917 7918 7919 7920
preprocessor perfmonitor: time 300 file /home/christ/Desktop/PhD/snort.stats
pktcnt 10000
preprocessor http_inspect: global iis_unicode_map unicode.map 1252
preprocessor http_inspect_server: server default \
   apache_whitespace no \
   ascii no \
        bare_byte no \
        chunk_length 500000 \
        flow_depth 1460 \
        directory no \
        double_decode no \
        iis_backslash no \
        iis_delimiter no \
        iis_unicode no \
        multi_slash no \
        non_strict \
        oversize_dir_length 500 \
        ports { 80 2301 3128 7777 7779 8000 8008 8028 8080 8180 8888 9999
} \
        u_encode yes \
        non_rfc_char { 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 } \
        webroot no

preprocessor rpc_decode: 111 32770 32771 32772 32773 32774 32775 32776
32777 32778 32779 no_alert_multiple_requests no_alert_large_fragments
no_alert_incomplete

preprocessor bo

preprocessor ftp_telnet: global encrypted_traffic yes check_encrypted
inspection_type stateful
preprocessor ftp_telnet_protocol: telnet \
   ayt_attack_thresh 20 \
   normalize ports { 23 } \
   detect_anomalies
```

```
preprocessor ftp_telnet_protocol: ftp server default \
    def_max_param_len 100 \
    ports { 21 2100 } \
    ftp_cmds { USER PASS ACCT CWD SDUP SMNT QUIT REIN PORT PASV TYPE
STRU MODE } \
    ftp_cmds { RETR STOR STOU APPE ALLO REST RNFR RNTO ABOR DELE
RMD MKD PWD } \
    ftp_cmds { LIST NLST SITE SYST STAT HELP NOOP } \
    ftp_cmds { AUTH ADAT PROT PBSZ CONF ENC } \
    ftp_cmds { FEAT OPTS CEL CMD MACB } \
    ftp_cmds { MDTM REST SIZE MLST MLSD } \
    ftp_cmds { XPWD XCWD XCUP XMKD XRMD TEST CLNT } \
    alt_max_param_len 0 { CDUP QUIT REIN PASV STOU ABOR PWD SYST
NOOP } \
    alt_max_param_len 100 { MDTM CEL XCWD SITE USER PASS REST DELE
RMD SYST TEST STAT MACB EPSV CLNT LPRT } \
    alt_max_param_len 200 { XMKD NLST ALLO STOU APPE RETR STOR CMD
RNFR HELP } \
    alt_max_param_len 256 { RNTO CWD } \
    alt_max_param_len 400 (Portokalidis and Bos) \
    alt_max_param_len 512 { SIZE } \
    chk_str_fmt { USER PASS ACCT CWD SDUP SMNT PORT TYPE STRU MODE }
\
    chk_str_fmt { RETR STOR STOU APPE ALLO REST RNFR RNTO DELE RMD
MKD } \
    chk_str_fmt { LIST NLST SITE SYST STAT HELP } \
    chk_str_fmt { AUTH ADAT PROT PBSZ CONF ENC } \
    chk_str_fmt { FEAT OPTS CEL CMD } \
    chk_str_fmt { MDTM REST SIZE MLST MLSD } \
    chk_str_fmt { XPWD XCWD XCUP XMKD XRMD TEST CLNT } \
    cmd_validity MODE < char ASBCZ > \
    cmd_validity STRU < char FRP > \
    cmd_validity ALLO < int [ char R int ] > \
    cmd_validity TYPE < { char AE [ char NTC ] | char I | char L [ number ] } >
\
    cmd_validity MDTM < [ date nnnnnnnnnnnnnn[.n[n[n]]] ] string > \
    cmd_validity PORT < host_port >
preprocessor ftp_telnet_protocol: ftp client default \
    max_resp_len 256 \
    bounce yes \
    telnet_cmds no

preprocessor smtp: ports { 25 587 691 } \
  inspection_type stateful \
  normalize cmds \
  normalize_cmds { EXPN VRFY RCPT } \
  alt_max_command_line_len 260 (Esmaili, Safavi-Naini et al.) \
```

```
  alt_max_command_line_len 300 { RCPT } \
  alt_max_command_line_len 500 { HELP HELO ETRN } \
  alt_max_command_line_len 255 { EXPN VRFY }

preprocessor ssh: server_ports { 22 } \
                max_client_bytes 19600 \
                max_encrypted_packets 20 \
                enable_respoverflow enable_ssh1crc32 \
                enable_srvoverflow enable_protomismatch

preprocessor dcerpc2: memcap 102400, events [co ]
preprocessor dcerpc2_server: default, policy WinXP, \
    detect [smb [139,445], tcp 135, udp 135, rpc-over-http-server 593], \
    autodetect [tcp 1025:, udp 1025:, rpc-over-http-server 1025:], \
    smb_max_chain 3

preprocessor dns: ports { 53 } enable_rdata_overflow

preprocessor ssl: ports { 443 465 563 636 989 992 993 994 995 7801 7702
7900 7901 7902 7903 7904 7905 7906 6907 7908 7909 7910 7911 7912 7913
7914 7915 7916 7917 7918 7919 7920 }, trustservers, noinspect_encrypted

output alert_csv: /home/christ/Desktop/alerts.csv default
output alert_full: alert

include /home/christ/Desktop/PhD/config/classification.config
include /home/christ/Desktop/PhD/config/reference.config

include $RULE_PATH/local.rules

include $RULE_PATH/exploit.rules
include $RULE_PATH/ftp.rules
include $RULE_PATH/telnet.rules
include $RULE_PATH/rpc.rules
include $RULE_PATH/rservices.rules
include $RULE_PATH/dos.rules
include $RULE_PATH/ddos.rules
include $RULE_PATH/dns.rules

include $RULE_PATH/web-cgi.rules
include $RULE_PATH/web-coldfusion.rules
include $RULE_PATH/web-iis.rules
include $RULE_PATH/web-frontpage.rules
include $RULE_PATH/web-misc.rules
include $RULE_PATH/web-client.rules
include $RULE_PATH/web-php.rules
```

```
include $RULE_PATH/sql.rules
include $RULE_PATH/x11.rules
include $RULE_PATH/netbios.rules
include $RULE_PATH/misc.rules
include $RULE_PATH/attack-responses.rules
include $RULE_PATH/oracle.rules
include $RULE_PATH/mysql.rules

include $RULE_PATH/smtp.rules
include $RULE_PATH/imap.rules
include $RULE_PATH/pop2.rules
include $RULE_PATH/pop3.rules

include $RULE_PATH/nntp.rules
include $RULE_PATH/backdoor.rules
include $RULE_PATH/snmp.rules
include $RULE_PATH/icmp.rules
include $RULE_PATH/tftp.rules
include $RULE_PATH/scan.rules
include $RULE_PATH/finger.rules
include $RULE_PATH/web-attacks.rules
include $RULE_PATH/shellcode.rules
include $RULE_PATH/policy.rules
include $RULE_PATH/info.rules
include $RULE_PATH/icmp-info.rules
include $RULE_PATH/virus.rules
include $RULE_PATH/chat.rules
include $RULE_PATH/multimedia.rules
include $RULE_PATH/p2p.rules
include $RULE_PATH/spyware-put.rules
include $RULE_PATH/specific-threats.rules
include $RULE_PATH/voip.rules
include $RULE_PATH/other-ids.rules
include $RULE_PATH/bad-traffic.rules

include /home/christ/Desktop/PhD/config/threshold.conf
```

APPENDIX F

PROFESSIONAL REVIEW

Appendix F. Professional Review

In order to gauge the view of other professionals it was decided to obtain independent review of the body of this thesis. Chapters 3, 4 and 5 were supplied to two consulting security architects to gain their views on the practicality of the proposed taxonomy and metrics. Specifically four questions were posed, as follows:

1) Do you agree with the premise that comparison of intrusion systems is difficult?
2) Does the taxonomy go some way towards addressing this difficulty?
3) Does the two new metrics help compare intrusion systems? and
4) Has the experimental program shown that the metrics are useful for comparing intrusion systems?

Unfortunately only one security architect could respond in time (Prowse 2013), the full content of which is repeated below:

"After reviewing the paper of Mr Chris Tucker and having worked as a professional security consultant and technical security architect with successful delivery of IDS and IPS systems for various clients my view of the paper is that it describes very well the issues that we see in day to day use and implementation of IDS and IPS systems and the difficulty that is often placed on the security professional when assisting clients with the choices for intrusion systems. Allied with the current crop of advanced persistent threats the paper demonstrates very clearly that the current approach to the way in which these system types are compared is not straight forward and requires to be

updated. The new taxonomy takes that step forward and brings with it known and demonstrable approaches and metrics for the measurement of these systems utilising techniques that have been widely used and are well understood from the radio and telegraphy sphere.

The proposed taxonomy allows for a greater comparison of systems and clearly demonstrates the difficulty in selection and use of the systems in a real world environment through the testing that was carried out in support of the paper. Given that the approach to security is never static and that in the current climate where customers and clients are expecting the "maximum bang for buck" taking this approach allows for the threat to be married more closely to the appropriate device type when the suggested new taxonomy model is used.

The adoption of the two new metrics should be seriously considered by suppliers of detection systems (used in the context of the paper to also include anti-virus systems) as a way of not only assisting in product improvement but one that also will allow for a client to make a more informed choice over the potential product or products that they may choose to apply to their environments. As stated previously experience has shown that clients typically now only rely on one type of IDS or IPS system within their environment so choice and demonstrable evidence in threat capture and analysis based on the updated matrix would allow for a more informed decision to be made.

The experimental program used in the paper would be useful for the comparison of intrusion systems and could be used in a wider context within the industry to compare product (possibly by SANs or similar), which would

allow for independent testing using the program to adopt and continue to expand the use of the new taxonomy and metrics.

Overall, the paper is well laid out and easy to read and understand for a professional in the information security field which a background in detection systems and poses some new and additional research options for the future. From a personal perspective the adoption of the updated taxonomy by the security industry should be seriously considered as it allows for a greater match to a wider threat profile and could if adopted allow a greater and more accurate choice of product and implementation of intrusion systems."

APPENDIX G

RESEARCH PAPERS

Appendix G. Research Papers

The following papers have been produced as part of the PhD research activities:

C. J. Tucker, S. M. Furnell, B. V. Ghita, and P. J. Brooke, "A new taxonomy for intrusion detection," in *International Networking Conference INC'06*. Plymouth, 2006 (Attached to this thesis);

C. J. Tucker, S. M. Furnell, B. V. Ghita, and P. J. Brooke, "A new taxonomy for comparing intrusion detection systems," *Internet Research*, vol. 17, pp. 88-98, 2007; (Attached to this thesis); and

C. J. Tucker, S. M. Furnell, B. V. Ghita, and P. J. Brooke, "The aggressive detection of network intrusions", submitted to Computers & Security.

A new taxonomy for intrusion detection

C.J.Tucker[1,2], S.M.Furnell[1], B.V.Ghita[1] and P.J.Brooke[3]

[1]Network Research Group, University of Plymouth, Plymouth, United Kingdom
[2]Stochastic Systems Limited, St Austell, Cornwall, United Kingdom
[3]School of Computing, University of Teesside, Middlesbrough, United Kingdom
e-mail: intrusion@stochastic.co.uk

Abstract

A new taxonomy is proposed in which the type of output and the data scale over which an intrusion system operates is used for classification. This taxonomy allows a graphical comparison of different intrusion systems to be undertaken in terms of their footprint on an intrusion matrix. It is proposed that quantitative comparison of systems can only be undertaken at points of overlap of their footprints and that overlap specific measures are needed for this comparison. New areas of application for intrusion systems are also discussed.

Keywords

Computer Security, Intrusion Detection, Taxonomy

1. Introduction

Intrusion detection has a long history, dating back to the work of Anderson (Anderson, 1980). Since then, various analysis techniques, ranging from support vector machines, through to data mining and expert systems, have been used as part of the detection engine (Mukkamala and Sung, 2003). Many complete systems have been constructed and operated on live computer systems (eg (Allen et al., 2005)). Despite over 25 years of research, the topic is still active, in part due to the rapid development of information processing systems and their vulnerabilities, but also due to fundamental difficulties in achieving an accurate declaration of an intrusion. Intrusion systems are noted for high false alarm rates and considerable research effort is still concentrated on finding effective intrusion, non-intrusion discriminants.

This paper proposes a new taxonomy which aims to improve the comparison of intrusion systems. The proposed taxonomy considers the different type of outputs that can be produced by intrusion systems, along with the type of information used to determine the intrusion, as the basis for their comparison.

2. Background

A number of taxonomies for intrusion detection have already been proposed. One of the earliest was undertaken by Debar and classified intrusion systems according to their detection method, behaviour on detection, audit source location, or usage frequency (Debar et al., 1999). This was later extended to include the detection paradigm, as either state- or transition-based (Debar et al., 2000). Axelsson offered an alternative taxonomy in terms of the detection

principle and operational aspects, such as whether operation is continuous or in batch mode (Axelsson, 2000).

Each of these taxonomies provides insight into the operation of intrusion systems and is a useful framework for identifying new research opportunities, but they are not a good basis for their comparison as they are based on the internal properties of such systems. A taxonomy based on the applicability of intrusion systems is a more fundamental comparison approach as it describes their use, rather than the details of their implementation.

Consider, for example, two network intrusion systems. System A is misuse-based whilst System B is anomaly-based. During a series of intrusion events both these systems will indicate the intrusion state of the network segment they are monitoring, possibly to different degrees of accuracy (i.e. their detection and false alarm rates). Much work has been published on the quantitative comparison of such systems (e.g. (Abouzakhar and Manson, 2004)), using analysis techniques such as receiver operating characteristic (ROC) or lift curves. However, in addition to the presence of an intrusion, System A will often indicate the type of attack and the exploit being used, on the basis of the specific signatures that are triggered. Comparing System A with System B via a ROC curve or confusion matrix will not include this important property of System A and thus is not a fair comparison method.

3. A new intrusion taxonomy

The taxonomy proposed in this paper is inspired by the work of Johnson in the formation of images (Johnson, 1958). He studied the ability of human operators to find and correctly classify objects within complex images. The objects were relatively small and thus the a priori probability that a specific area of the image contained an object was very low, a situation analogous to intrusion events within a background of normal network or host activity (Axelsson, 1999). Johnson defined the following types of operator tasks (amongst others):

- **Detection** – the ability to say that something of interest is present in the image.
- **Recognition** – the ability to determine the class of object present, such as a car or aircraft.
- **Identification** – the ability to determine the type of object present, such as the make of car or the type of aircraft.

The most important aspect of Johnson's work was the definition of minimum criteria necessary for successful completion of the above tasks. Using similar task definitions as a starting point, this study proposes to divide the output of intrusion systems into one of 5 categories, with the first 3 loosely in line with Johnson, as follows:

- **Detection** – in which the system outputs an indication of a state change within a network or host. There is no classification of the nature of the change, apart for the assumption that this indicates the occurrence of a possible intrusion. The principal use of such systems is for data rate reduction so that other systems (either automated or human) can investigate further.
- **Recognition** – in which the intrusion systems are capable of declaring the type of attack, such as Distributed Denial of Service (DDoS), reconnaissance, or User to Root (U2R).

- **Identification** – in which the system is capable of declaring the exploits used to achieve the intrusion, such as buffer overflow or an application-specific vulnerability.
- **Confirmation** – in which the attack plan is deduced, allowing attack-specific countermeasures to be deployed rather than coarse measures, such as disconnection of the internet access or isolation of key business servers.
- **Prosecution** – in which evidential quality data is generated identifying the originator of the intrusion.

As an example of the use of this taxonomy consider a simple anomaly intrusion system comparing the utilised network bandwidth with historical values. Such a system would be categorised as an intrusion detection system. It would be able to declare that something unusual is happening within the network but declaring with any certainty that the anomaly was caused by an intruder is not likely to be achievable.

As another example consider a Snort intrusion system operating on a single network segment (Roesch, 1999). When a rule is triggered and an alert declared, there is considerable attack-related information available. Often, rules are created to alert when the signatures of specific attacks are present. Thus, when such a rule has been triggered, the intrusion system can identify the exploit being used. In this respect Snort is acting as an intrusion identification system.

In addition to considering the output from an intrusion system, further insight can be achieved from an analysis of the data scale over which the system is operating. In modern network systems four data scales can be considered:

- **File** – monitoring the status of individual files for unauthorised access or change.
- **Host** –monitoring the applications running on and the behaviour of an individual host.
- **Network** – monitoring the packets exchanged between hosts, servers and other network devices to assert the presence of an intrusion.
- **Enterprise** – monitoring traffic originating from trusted sources of an organisation which operate in the presence of other, less trusted data sources.

The File, Host and Network data scales have been used in other studies (e.g. (Bace and Mell, 2001)). The separation of Network data scale into two sections, as introduced by this paper, is believed to be a novel concept. The principal difference between the Network and Enterprise data scales is the mixing of trusted and untrusted data streams within the same network segment. This is most often encountered in virtual private networks (VPN) between an office location of an organisation and its remote staff or trusted partners, via the Internet. VPNs are separated from the untrusted data streams using encryption schemes and well-known protocols. However, this separation may become subject to the same technology, policy or configuration vulnerabilities as other parts of the information processing system. Therefore, it is likely that an individual responsible for a secure network would want to know that their VPN communications were subject to intrusion attempts. Intrusion systems therefore need to extend their data scale applicability to include the Enterprise. This is a technically challenging problem.

4. The application of the taxonomy

This taxonomy can be applied in a number of ways. The remainder of this paper will examine its use to create an intrusion footprint on a grid or matrix formed from the output type and data scale elements of the taxonomy. The use of this footprint for comparison of systems will then be shown.

4.1 Intrusion matrix

The combination of intrusion output type and data scale can be shown as an intrusion matrix, as in figure 1.

	FILES	HOST	NETWORK	ENTERPRISE
PROSECUTION	Protective Monitoring, Secure Data Vaults	Protective Monitoring, Secure Data Vault	Anti-spoofing, Trust Management	-
CONFIRMATION	A Priori Assessment	A Priori Assessment	AI Techniques	-
IDENTIFICATION	Malware Signatures	Malware Signatures	Exploit Signatures	-
RECOGNITION	File Hashes	Malware Signatures, Resource Anomalies	Traffic, Protocol or User Anomalies	-
DETECTION	File Hashes	Sys Calls, Registry Use, Resource Anomalies	Traffic, Protocol or User Anomalies	Quantum Cryptography

Figure 1: Intrusion System Taxonomy Matrix

Also shown in figure 1 are some of the techniques that can be applied within a particular output type and data scale. For example, malware signatures or resource anomalies can be used in intrusion recognition systems operating at the Host data scale. Much of this matrix is covered with techniques that have been extensively studied. Of particular note is the absence of techniques at the Enterprise data scale.

4.2 Intrusion system footprint

The intrusion matrix can be used to plot a footprint for different intrusion systems. The footprints are determined from an analysis of the intrusion system outputs to determine which of the 5 output categories the system is capable of producing and what data scale is used to create the output. For example, figure 2 shows the footprints of a number of different intrusion paradigms. Figure 2a shows the footprint of a representative anti-virus software (AVS) package. They typically include both virus-specific signatures and heuristics that respond to anomalous behaviours. This means they operate from the Detection to the Identification output types. Since the attack plan can often be determined by reverse

engineering of the virus, AVS packages can also be considered to operate at the Confirmation output type.

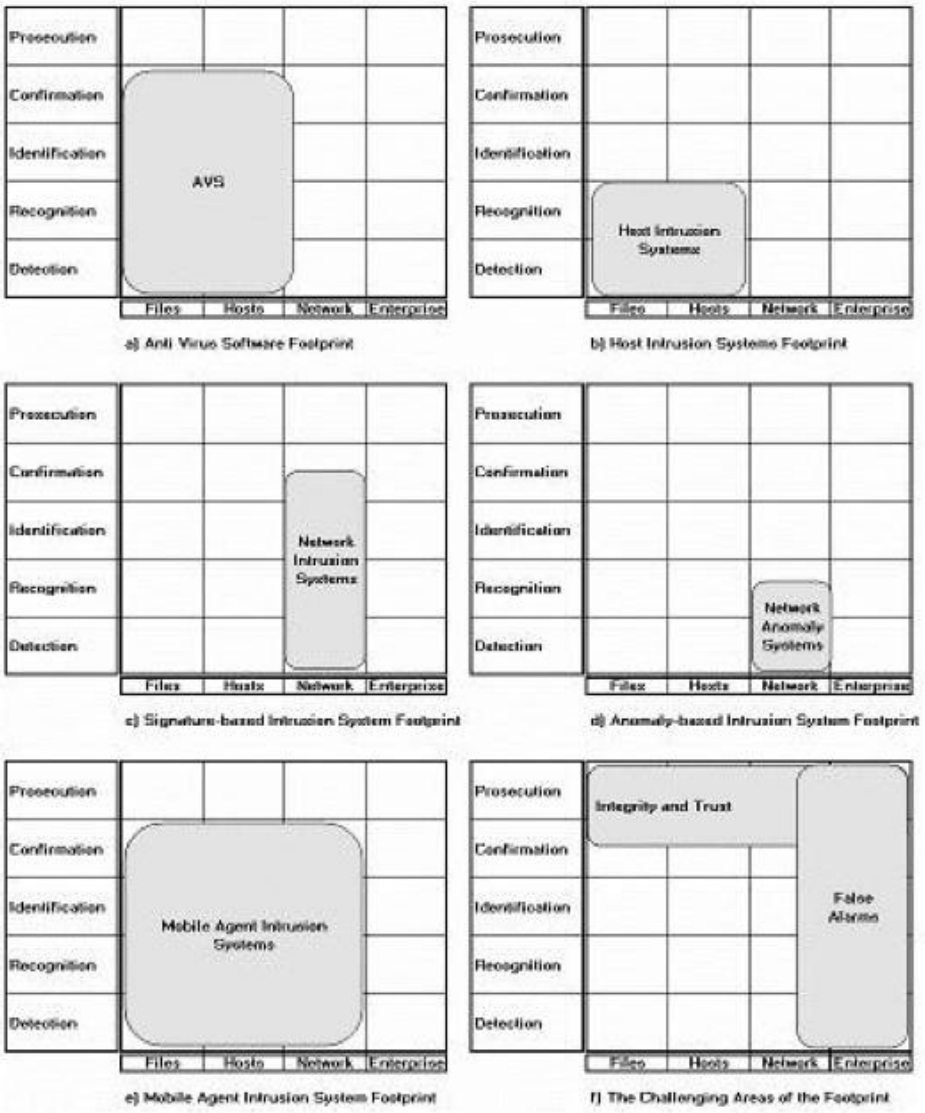


Figure 2: Intrusion System Footprints

A footprint of a host-based intrusion system is shown in figure 2b. To create this footprint it was assumed that anomaly techniques are applied and therefore the intrusion system is only capable of Detection or Recognition. Confirmation, or the determination of the specific exploit or vulnerability used (Identification), are unlikely to be achievable with confidence when using an anomaly based system. Host-based intrusion systems using signature techniques would be expected to operate at the Identification and Confirmation levels, depending on the discrimination capabilities of the signatures.

Figures 2c and 2d show network-based intrusion systems using signature and anomaly detection respectively. These figures highlight the principal differences to be at the higher output types of Identification and Confirmation. Snort is a typical example of a signature

based intrusion system. On its own it is unable to perform the plan determination required for full Confirmation. However, when multiple Snort sensors are deployed at strategic parts of a network, it is possible to determine an attack plan from the patterns of signatures that are triggered. An additional module would be required to integrate the information and hence determine the plan. Hence, the Confirmation output type is shown partially covered by the footprint.

Figure 2e is the most interesting, and shows the extensive footprint that could be achieved by intrusion systems based on mobile agents. On the assumption that mobile agents could be created to examine the status of files, applications running on a host, and packets on the local network segment, they offer the widest range of data scales of any other technique. Also, their payload could include integrated anomaly and signature based techniques, and when combined with a communications capability this could give them the potential to provide output types up to Confirmation. It may even be possible that techniques for Enterprise data scales and Prosecution could be integrated as they become available.

Finally, figure 2f shows the current challenges faced by intrusion systems. The Prosecution output type requires high integrity information to be gathered and secured from change. Whilst this is a common requirement in secure systems it must be achieved to the levels necessary to allow criminal prosecution, within a system that has intruders present. For the Enterprise data scale, the technology challenge appears to be the development of discriminants that will separate intrusion and non-intrusion events in mixed-trust data flows.

4.3 Comparison of intrusion systems

The intrusion matrix can be used to provide a comparison between systems. A qualitative comparison can be made by examining the footprint of each system. Large footprints are likely to represent systems which provide a broader range of applicability and a wider range of output information during an intrusion. Small footprints would be typical for systems which are very specific in their application.

A more quantitative comparison can be made by examining the performance of systems where their footprints overlap. Each element of the intrusion matrix is accompanied by a set of performance metrics relevant to the output data type. These performance metrics could include false alarm rates, intrusion probabilities, or confusion matrices measured in such a way as to be appropriate to the position within the intrusion matrix. As an example consider a single element within the intrusion matrix, say the (Network, Identification) element. If the footprint of two intrusion systems overlap on this element then performance metrics relevant to Identification should be calculated for the 2 systems. The probability of identification could be determined as a function of the false alarm rate, to produce Identification ROC curves. Examination of the ROC curves at this overlap point within the intrusion matrix would allow comparison of the systems in the role of intrusion identification. A fair comparison would require the examination of performance metrics at all points of overlap on the intrusion matrix.

Some of the elements of the intrusion matrix presented have been extensively studied and can be considered commercial successes. For example, AVS packages are very successful at providing confident alerts at the Files and Host data scales (Post and Kagan, 1998). Such software can be very specific, identifying the virus and hence, by implication the "plan" of the originator of the virus. Heuristic algorithms can provide a degree of detection capability in

which the AVS indicates that there is a virus present but is not specific about its type. AVS packages are also well known to provide a high alert probability with a low false alarm rate. Thus a large area of this matrix can be achieved with very high performance.

Meanwhile, some of the elements of the intrusion matrix are poorly understood at this time. Effective techniques at the Enterprise data scale are rare and of limited applicability. This applies at any of the intrusion output levels. The trusted data stream may be present with untrusted streams and on untrusted network equipment (e.g. Internet backbone routers). Intrusion systems are unlikely to be able to operate outside of the trusted systems of the enterprise, leaving Enterprise scale intrusion systems to rely on remote diagnosis of intrusion behaviour.

It can therefore be seen that there are 3 aspects of the intrusion matrix that are important in determining the performance of an intrusion system, namely:

a) The number of elements of the matrix that an individual system footprint covers as this can indicate the applicability of the system.
b) The position of the elements of the footprint within the intrusion matrix, as some element positions present an inherently challenge to achieve high performance
c) Only the elements that overlap are of any significance in the quantitative comparison of intrusion system.

5. Conclusions and further work

This paper proposed a novel taxonomy for intrusion systems, based on the type of information the system is capable of providing, as well as the data scale over which it operates. It allows a graphical comparison of systems to be undertaken, using their footprint on an intrusion matrix constructed from the output capabilities and data scale. A more precise, quantitative comparison can then be undertaken at overlapping elements within the footprint, using metrics specific to the type of output. Thus ROC curves based on Probability of Recognition versus Recognition False Alarms would be created to compare systems that have a footprint overlap at the Recognition output type.

Finally, the inclusion of AVS within this taxonomy opens the challenging and interesting alternative of placing intrusion systems on a more theoretical basis. The work of Cohen (Cohen, 1987) has already established theoretical limits on the detectability of viruses, proving that no algorithm can perfectly detect all possible viruses. More recently Li has proposed a theoretical basis for intrusion, but this work has yet to reveal any useful conclusions (Li et al., 2005). It is hoped that this taxonomy will build on this theoretical basis and lead to a better understanding of the limits of performance for intrusion systems, as well as providing an improved framework for their comparison.

6. References

Abouzakhar, N. S. & Manson, G. A. (2004), "Evaluation of intelligent intrusion detection Models", *International Journal of Digital Evidence,* 3.

Allen, W. H., Marin, G. A. & Rivera, L. A. (2005), "Automated detection of malicious reconnaissance to enhance network security", *SoutheastCon, 2005. Proceedings. IEEE*

Anderson, J. (1980), "Computer security, threat monitoring and surveillance", Fort Washington PA, James P Anderson Co.

Axelsson, S. (1999), "Base-rate fallacy and its implications for the difficulty of intrusion detection", *Proceedings of the 1999 6th ACM Conference on Computer and Communications Security (ACM CCS), Nov 2-Nov 4 1999*, Singapore, Singapore

Axelsson, S. (2000), "Intrusion detection systems: A survey and taxonomy", Department of Computer Engineering, Chalmers University.

Bace, R. & Mell, P. (2001), "Intrusion detection systems", *NIST Special Publication on Intrusion Detection System*.

Cohen, F. (1987), "Computer viruses: Theory and experiments", *Computers and Security,* 6, 22-35.

Debar, H., Dacier, M. & Wespi, A. (1999), "Towards a taxonomy of intrusion-detection systems", *Computer Networks-the International Journal of Computer and Telecommunications Networking,* 31, 805-822.

Debar, H., Dacier, M. & Wespi, A. (2000), "A revised taxonomy for intrusion-detection systems", *Annales Des Telecommunications-Annals of Telecommunications,* 55, 361-378.

Johnson, J. (1958), "Analysis of image forming systems", *Proceedings of the Image Intensifier Symposium*, US Army Engineering Research Development Laboratories, Fort Belvoir, USA

Li, Z., Das, A. & Zhou, J. (2005), "Theoretical basis for intrusion detection", *Systems, Man and Cybernetics (SMC) Information Assurance Workshop, 2005. Proceedings from the Sixth Annual IEEE*

Mukkamala, S. & Sung, A. H. (2003), "A comparative study of techniques for intrusion detection", *Tools with Artificial Intelligence, 2003. Proceedings. 15th IEEE International Conference on*,

Post, G. & Kagan, A. (1998), "The use and effectiveness of anti-virus software", *Computers & Security,* 17, 589-599.

Roesch, M. (1999), "Snort - Lightweight intrusion detection for networks", *Proceedings of USENIX 13th Systems Administration Conference (LISA '99)*, Berkeley, CA

A new taxonomy for comparing intrusion detection systems

C.J.Tucker[1,2], S.M.Furnell[1], B.V.Ghita[1] and P.J.Brooke[3]

[1]Network Research Group, School of Computing, Communications & Electronics, University of Plymouth, Plymouth, United Kingdom
[2]Stochastic Systems Limited, St Austell, Cornwall, United Kingdom
[3]School of Computing, University of Teesside, Middlesbrough, United Kingdom

Keywords Computer Security, Intrusion Detection, Taxonomy

Paper type Research Paper

Abstract

Purpose – To propose a new taxonomy for intrusion detection systems as a way of generating further research topics focussed on improving intrusion system performance.

Approach – Intrusion systems are characterised by the type of output they are capable of producing, such as intrusion/non-intrusion declarations, through to intrusion plan determination. The output type is combined with the data scale used to undertake the intrusion determination, to produce a two-dimensional intrusion matrix.

Findings – Different approaches to intrusion detection can produce different footprints on the intrusion matrix. Qualitative comparison of systems can be

undertaken by examining the area covered within the footprint and the footprint overlap between systems. Quantitative comparison can be achieved in the areas of overlap.

Research Implications - The comparison of systems based on their footprint on the intrusion matrix may allow a deeper understanding of the limits of performance to be developed. The separation of what was previously understood as "detection" into the three areas of Detection, Recognition and Identification may provide further impetus for the development of a theoretical framework for intrusion systems.

Practical Implications – The intrusion matrix can be divided into areas in which the achievement of arbitrarily high performance is relatively easily achievable. Other areas within the matrix, such as the Prosecution and Enterprise regions, present significant practical difficulties and therefore are opportunities for further research.

Originality - The use of a taxonomy based on the type of output produced by an intrusion system is new to this paper, as is the combination with data scale to produce an intrusion matrix. The recognition that the network data scale should also be split to differentiate trusted and untrusted networks is new and presents challenging opportunities for further research topics.

Introduction

Intrusion detection has a long history, dating back to the work of Anderson (Anderson, 1980). Since then, various discrimination techniques, ranging from support vector machines, through to data mining and expert systems, have been used as part of the detection engine (Mukkamala and Sung, 2003). Many complete systems have been constructed and operated on live computer systems (for example (Allen et al., 2005)). However, despite over 25 years of research, the topic is still active, in part due to the rapid development of information processing systems and the consequent discovery of new vulnerabilities, but also due to fundamental difficulties in achieving an accurate declaration of an intrusion. Intrusion systems are noted for high false alarm rates and considerable research effort is still concentrated on finding effective intrusion, non-intrusion discriminants.

This paper proposes a new taxonomy which aims to improve the comparison of intrusion systems. Taxonomies are an important aspect of the analysis of systems. They can act as a description, providing order to the subject and an explanation of observed phenomena. More importantly, they can provide insights through the identification of gaps. Such insights often identify new areas of research and this was the motivation for developing the taxonomy described in this paper.

The proposed taxonomy considers the different type of outputs that can be produced by intrusion systems, along with the type of information used to determine the intrusion, as the basis for their comparison. A graphical combination of these parameters is proposed against which intrusion systems can be qualitatively and quantitatively compared.

First, the background to the problem of comparing intrusion detection systems is presented in terms of other taxonomies. The difficulties of comparing an anomaly based system with a network intrusion system using signatures are discussed. Then the new taxonomy is presented in terms of two parameters, namely the output type and the data scale over which they operate. The bulk of this paper addresses the application of the taxonomy by first introducing the concept of an intrusion matrix, to combine the two parameters, and then the concept of intrusion footprints to plot the capabilities of individual intrusion systems. The use of the taxonomy to compare different systems is then discussed before the potential for further theoretical analysis is described.

Background

A number of taxonomies for intrusion detection have already been proposed. One of the earliest was undertaken by Debar at al and classified intrusion systems according to their detection method, behaviour on detection, audit source location, or usage frequency (Debar et al., 1999). This was later extended to include the detection paradigm, as either state- or transition-based (Debar et al., 2000). Axelsson offered an alternative taxonomy in terms of the detection principle and operational aspects, such as whether operation is continuous or in batch mode (Axelsson, 2000). More recently taxonomies have been developed that classify intrusion systems according

to the attack stage they can declare intrusions, such as pre-attack, real-time or post attack (Lukatsky, 2002).

Each of these taxonomies provides insight into the operation of intrusion systems and is a useful framework for identifying new research opportunities. However, they are not a good basis for their comparison as they use the internal properties of such systems for classification. A taxonomy based on the applicability of intrusion systems is a more fundamental comparison approach as it describes their use, rather than the details of their implementation.

Consider, for example, two network intrusion systems. System A is misuse-based whilst System B is anomaly-based. During a series of intrusion events both these systems will indicate the intrusion state of the network segment they are monitoring, possibly to different degrees of accuracy (that is, their detection and false alarm rates). Much work has been published on the quantitative comparison of such systems (for example (Abouzakhar and Manson, 2004)), using analysis techniques such as receiver operating characteristic (ROC) or lift curves. However, in addition to the presence of an intrusion, System A will often indicate the type of attack and the exploit being used, on the basis of the specific signatures that are triggered. Comparing System A with System B via a ROC curve or confusion matrix will not include this important property of System A and thus is not a fair comparison method.

Consider also the use of the output from these two systems. If both systems are providing alerts to network support staff the actions that are likely to be taken are different. System A will identify the network peers involved in the suspected intrusion behaviour as well as the nature of the attack, allowing support staff to take specific action quickly. Support staff using System B may need to undertake further investigations before the information necessary to stop the intrusion behaviour is derived. In summary, each of these approaches to intrusion detection makes differing demands on the systems that use the information they provide and therefore comparison techniques should include this in their assessment.

The new intrusion taxonomy

The taxonomy proposed in this paper is inspired by the work of Johnson in the formation of images (Johnson, 1958). He studied the ability of human operators to find and correctly classify objects within complex images. The objects were relatively small and thus the a priori probability that a specific area of the image contained an object was very low, a situation analogous to intrusion events within a background of normal network or host activity (Axelsson, 1999). Johnson defined the following types of operator tasks (amongst others):

- **Detection** – the ability to say that something of interest is present in an image;
- **Recognition** – the ability to determine the class of object present, such as a car or aircraft; and
- **Identification** – the ability to determine the type of object present, such as the make of car or the type of aircraft.

The most important aspect of Johnson's work was the definition of minimum criteria necessary for successful completion of the above tasks. Using similar task definitions as a starting point, this study proposes to divide the output of intrusion systems into one of 5 categories, with the first 3 loosely in line with Johnson, as follows:

- **Detection** – in which the system outputs an indication of a state change within a network or host. There is no classification of the nature of the change, apart for the assumption that this indicates the occurrence of a possible intrusion. The principal use of such systems is for data rate reduction so that other systems (either automated or human) can investigate further;
- **Recognition** – in which the intrusion systems are capable of declaring the type of attack, such as Distributed Denial of Service (DDoS), reconnaissance, or User to Root (U2R);
- **Identification** – in which the system is capable of declaring the exploits used to achieve the intrusion, such as buffer overflow or an application-specific vulnerability;

- **Confirmation** – in which the attack plan is deduced, allowing attack-specific countermeasures to be deployed rather than coarse measures, such as disconnection of the internet access or isolation of key business servers; and
- **Prosecution** – in which evidential quality data is generated identifying the originator of the intrusion.

As an example of the use of this taxonomy consider a simple anomaly intrusion system comparing the utilised network bandwidth with historical values. Such a system would be categorised as an intrusion detection system. It would be able to declare that something unusual is happening within the network but declaring with that the anomaly was caused by an intruder is not likely to be achievable to an arbitrarily high accuracy.

As another example consider a Snort intrusion system operating on a single network segment (Roesch, 1999). When a rule is triggered and an alert declared, there is considerable attack-related information available. Often, rules are created to alert when the signatures of specific attacks are present. Thus, when such a rule has been triggered, the intrusion system can identify the exploit being used, as well as the network peers involved. Within the taxonomy proposed here Snort is acting as an intrusion identification system.

In addition to considering the output from an intrusion system, further insight can be achieved from an analysis of the data scale over which the system is operating. In modern computer systems four data scales can be considered:

- **File** – monitoring the status of individual files for unauthorised access or change;
- **Host** –monitoring the applications running on and the behaviour of an individual host;
- **Network** – monitoring the packets exchanged between hosts, servers and other network devices to assert the presence of an intrusion; and
- **Enterprise** – monitoring traffic originating from trusted sources of an organisation which operate in the presence of other, less trusted data sources.

The File, Host and Network data scales have been used in other studies (for example, (Bace and Mell, 2001)). The separation of Network data scale into two sections, as introduced by this paper, is believed to be a novel concept. The principal difference between the Network and Enterprise data scales is the mixing of trusted and untrusted data streams within the same network segment. This is most often encountered in virtual private networks (VPN) between an office location of an organisation and its remote staff or trusted partners, via the Internet. VPNs are separated from the untrusted data streams using encryption schemes and well-known protocols. However, this separation may become subject to the same technology, policy or configuration vulnerabilities as other parts of the information processing system. Therefore, it is likely that security staff will want to know when their VPN communications are subject to intrusion attempts. Intrusion systems therefore need to extend their data scale applicability to include the Enterprise. This is a technically challenging problem.

The application of the taxonomy

This taxonomy can be applied in a number of ways. The remainder of this paper will examine its use to create an intrusion footprint on a grid or matrix formed from the output type and data scale elements of the taxonomy. The use of this footprint for comparison of systems will then be shown.

Intrusion matrix

The combination of intrusion output type and data scale can be shown as an intrusion matrix, as in figure 1. Also shown in figure 1 are some of the techniques that can be applied within a particular output type and data scale. For example, malware signatures or resource anomalies can be used in intrusion recognition systems operating at the Host data scale. Much of this matrix is covered with techniques that have been extensively studied. Of particular note is the absence of practical techniques at the Enterprise data scale.

	FILES	HOST	NETWORK	ENTERPRISE
PROSECUTION	Protective Monitoring, Secure Data Vaults	Protective Monitoring, Secure Data Vault	Anti-spoofing, Trust Management	-
CONFIRMATION	A Priori Assessment	A Priori Assessment	AI Techniques	-
IDENTIFICATION	Malware Signatures	Malware Signatures	Exploit Signatures	-
RECOGNITION	File Hashes	Malware Signatures, Resource Anomalies	Traffic, Protocol or User Anomalies	-
DETECTION	File Hashes	Sys Calls, Registry Use, Resource Anomalies	Traffic, Protocol or User Anomalies	Quantum Cryptography

Figure 1: Intrusion System Taxonomy Matrix

Intrusion system footprint

The intrusion matrix can be used to plot a footprint for different intrusion systems. The footprints are determined from an analysis of the intrusion system outputs to determine which of the 5 output categories the system is capable of producing and what data scale is used to create the output. For example, figure 2 shows the footprints of a number of different intrusion paradigms. Figure 2a shows the footprint of a representative anti-virus software (AVS) package. They typically include both virus-specific signatures and heuristics that respond to anomalous behaviours. This means they operate from the Detection to the Identification output types. Since the attack plan can often be determined by reverse engineering of the virus, AVS packages can also be considered to operate at the Confirmation output type.

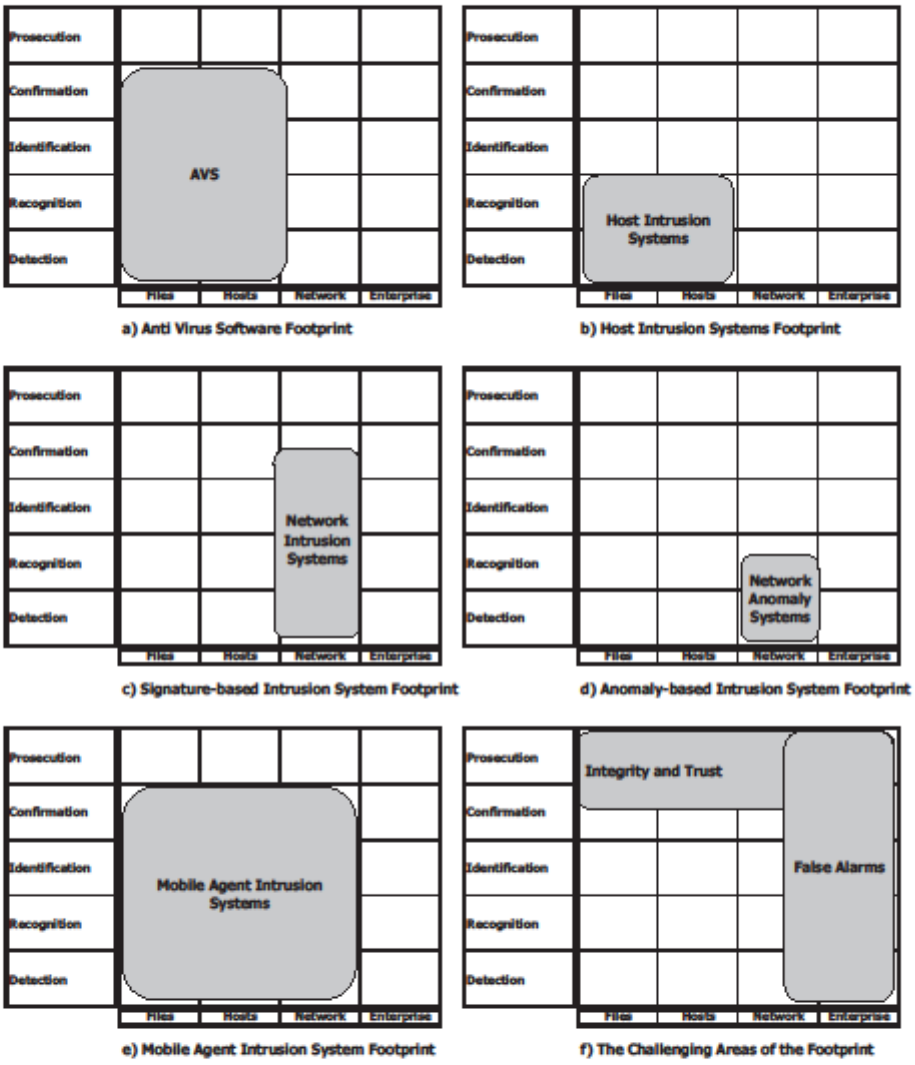


Figure 2: Intrusion System Footprints

A footprint of a host-based intrusion system is shown in figure 2b. To create this footprint it was assumed that anomaly techniques are applied and therefore the intrusion system is only capable of Detection or Recognition. Confirmation, or the determination of the specific exploit or vulnerability used (Identification), are unlikely to be achievable with confidence when using an anomaly based system. Host-based intrusion systems using signature techniques would be expected to operate at the Identification and Confirmation levels, depending on the discrimination capabilities of the signatures.

Figures 2c and 2d show network-based intrusion systems using signature and anomaly detection respectively. These figures highlight the principal differences to be at the higher output types of Identification and Confirmation. Snort is a typical example of a signature based intrusion system. On its own it is unable to perform the plan determination required for full Confirmation. However, when multiple Snort sensors are deployed at strategic parts of a network, it is possible to determine an attack plan from the patterns of signatures that are triggered. An additional module would be required to integrate the information and determine the plan. Hence, the Confirmation output type is shown partially covered by the footprint.

Figure 2e is the most interesting, and shows the extensive footprint that could be achieved by intrusion systems based on mobile agents. On the assumption that mobile agents could be created to examine the status of files, applications running on a host, and packets on local network segments, they offer the widest range of data scales of any other technique. Also, their payload could include integrated anomaly and signature based techniques, and when combined with a communications capability this could give them the potential to provide output types up to Confirmation. It may even be possible that techniques for Enterprise data scales and Prosecution could be integrated as they become available.

Finally, figure 2f shows the current challenges faced by intrusion systems. The Prosecution output type requires high integrity information to be gathered and secured from change. Whilst this is a common requirement in secure systems it must be achieved to the levels necessary to allow criminal prosecution, within a system

that has intruders present (Sommer, 1999). For the Enterprise data scale, the technology challenge appears to be the development of discriminants that will separate intrusion and non-intrusion events in mixed-trust data flows. Such data flows will often be occurring on equipments not owned by the enterprise and therefore the ability to provide local monitoring of the network will be limited.

Comparison of intrusion systems

The intrusion matrix can be used to provide a comparison between systems. A qualitative comparison can be made by examining the footprint of each system. Large footprints are likely to represent systems which provide a broader range of applicability and a wider range of output information during an intrusion. Small footprints would be typical for systems which are very specific in their application.

A more quantitative comparison can be made by examining the performance of systems where their footprints overlap. Each element of the intrusion matrix is accompanied by a set of performance metrics relevant to the output data type. These performance metrics could include false alarm rates, intrusion probabilities, or confusion matrices measured in such a way as to be appropriate to the position within the intrusion matrix. As an example consider a single element within the intrusion matrix, say the (Network, Identification) element. If the footprint of two intrusion systems overlap on this element then performance metrics relevant to Identification should be calculated for the 2 systems. The probability of identification could be determined as a function of the false alarm rate, to produce Identification ROC curves. Examination of the ROC curves at this overlap point within the intrusion matrix would allow comparison of the systems in the role of intrusion identification. A fair comparison would require the examination of performance metrics at all points of overlap on the intrusion matrix as well as a recognition of the additional capabilities offered at points where they do not overlap.

Some of the elements of the intrusion matrix presented have been extensively studied and can be considered commercial successes. For example, AVS packages are very successful at providing confident alerts at the Files and Host data scales (Post and Kagan, 1998). Such software can be very specific, identifying the virus and

hence, by implication the "plan" of the originator of the virus. Heuristic algorithms can provide a degree of detection capability in which the AVS indicates that there is a virus present but is not specific about its type. AVS packages are also well known to provide a high alert probability with a low false alarm rate. Thus a large area of this matrix can be achieved with very high performance.

Meanwhile, some of the elements of the intrusion matrix are poorly understood at this time. Effective techniques at the Enterprise data scale are rare and of limited applicability. This applies at any of the intrusion output capabilities. The trusted data stream may be present with untrusted streams and on untrusted network equipment (for example, Internet backbone routers). Current intrusion systems are not able to operate outside of the trusted systems of the enterprise, leaving Enterprise scale intrusion systems to rely on remote diagnosis of intrusion behaviour.

It can therefore be seen that there are 3 aspects of the intrusion matrix that provide insight to the performance of an intrusion system and should be considered when comparing systems, namely:

a) The number of elements of the matrix that an individual system footprint covers as this can indicate its applicability;
b) The position of the elements of the footprint within the intrusion matrix, as some element positions present a significant challenge to the achievement of high performance; and
c) Only the elements that overlap are of any significance in the direct quantitative comparison of intrusion systems.

Relationship with other definitions of intrusion

One of the earliest definitions of intrusion was from Amoroso. He defined intrusion detection as "the process of identifying and responding to malicious activity targeted at computing and networking resources" (Amoroso, 1998). In the same year Ptacek and Newsham defined intrusion as "unauthorized usage of or misuse of a computer system (Ptacek and Newsham, 1998) whilst Alessandri et al defined intrusion as "a malicious activity threatening the security policy that leads to a security failure, that is to a security policy violation" (Alessandri et al., 2001). More recently many

researchers have used the definition of Bace and Mell in which intrusion is defined as "attempts to compromise the confidentiality, integrity, availability, or to bypass the security mechanisms of a computer or network" (Bace and Mell, 2001).

For the remainder of this paper we will use a simple definition based on Alessandri, without the restriction of malicious intent. Therefore we consider intrusion to be defined as "an activity that leads to the violation of the security policy of a computer system". Further insight can be gained by considering the relationship of this definition with the definitions of detection, recognition and identification presented earlier.

In this context, intrusion detection can be seen as the declaration that the security policy has been violated, but the specific clause that has been violated is not declared. Intrusion recognition systems are able to declare which clauses or subsets of clauses have been violated. Intrusion identification systems are able to declare which clauses or subsets have been violated, as well as declaring the way in which they have been violated.

The above discussion can be used as the basis of a mathematical model of the intrusion declaration process, potentially allowing the theoretical limits to be determined in the same manner as Johnson's work for imaging systems. Also the inclusion of AVS within this taxonomy opens the challenging and interesting option of building on the theoretical work already published in this area. The work of Cohen (Cohen, 1987) has already established theoretical limits on the detectability of viruses, proving that no algorithm can perfectly detect all possible viruses. More recently Li et al have proposed a theoretical basis for intrusion, but this work has yet to reveal any useful conclusions (Li et al., 2005). It is hoped that this taxonomy will build on this theoretical basis and lead to a better understanding of the limits of performance for intrusion systems, as well as providing an improved framework for their comparison.

Conclusions and further work

This paper proposed a novel taxonomy for intrusion systems, based on the type of information the system is capable of providing, as well as the data scale over which it operates. It allows a graphical comparison of systems to be undertaken, using their footprint on an intrusion matrix constructed from the output capabilities and data scale. A more precise, quantitative comparison can then be undertaken at overlapping elements within the footprint, using metrics specific to the type of output. Individual ROC curves can be produced and compared for the intrusion systems operating as detection, recognition or identification systems, depending on their overlap within the intrusion matrix. Further work is necessary to determine how to combine the many differing performance metrics that can be generated for this approach, into a single definition of what constitutes the "best" intrusion system for a given problem.

The challenging areas of the intrusion matrix have been identified as the Enterprise data scale, for all intrusion output capabilities and the Prosecution output capability, for all data scales. Although careful design techniques can allow Prosecution systems to be proposed and constructed, the techniques available at the Enterprise data scale are sparse and present significant technical and legal difficulties.

Finally, further work is still to be undertaken in establishing a useful theoretical framework for intrusion systems. A variety of definitions for intrusion are used by the many researchers active in this field. A logical start is therefore to agree on a single definition and build a theory around it. The expansion of what was thought of as "detection" into 3 separate and distinct activities presented in this paper may provide further impetus for this goal.

References

ABOUZAKHAR, N. S. & MANSON, G. A. 2004. Evaluation of intelligent intrusion detection models. *International Journal of Digital Evidence,* 3.

ALESSANDRI, D., CACHIN, C., DACIER, M., DEAK, O., JULISCH, K., RANDELL, B., RIORDAN, J., TSCHARNER, WESPI, A. & WUEST, C. 2001. Towards a taxonomy of intrusion detection systems and attacks. IBM Research, Zurich Research Laboratory.

ALLEN, W. H., MARIN, G. A. & RIVERA, L. A. Automated detection of malicious reconnaissance to enhance network security. SoutheastCon, 2005. Proceedings. IEEE, 2005. 450-454.

AMOROSO, E. G. 1998. *Intrusion Detection : An introduction to Internet surveillance, correlation, traps, race back, and response,* Sparta, N.J., Intrusion.Net Books.

ANDERSON, J. 1980. Computer security, threat monitoring and surveillance. Fort Washington PA: James P Anderson Co.

AXELSSON, S. Base-rate fallacy and its implications for the difficulty of intrusion detection. Proceedings of the 1999 6th ACM Conference on Computer and Communications Security (ACM CCS), Nov 2-Nov 4 1999, 1999 Singapore, Singapore. ACM, New York, NY, USA, 1-7.

AXELSSON, S. 2000. Intrusion detection systems: A survey and taxonomy. Department of Computer Engineering, Chalmers University.

BACE, R. & MELL, P. 2001. Intrusion detection systems. *NIST Special Publication on Intrusion Detection System.*

COHEN, F. 1987. Computer viruses: Theory and experiments. *Computers and Security,* 6, 22-35.

DEBAR, H., DACIER, M. & WESPI, A. 1999. Towards a taxonomy of intrusion-detection systems. *Computer Networks-the International Journal of Computer and Telecommunications Networking,* 31, 805-822.

DEBAR, H., DACIER, M. & WESPI, A. 2000. A revised taxonomy for intrusion-detection systems. *Annales Des Telecommunications-Annals of Telecommunications,* 55, 361-378.

JOHNSON, J. Analysis of image forming systems. Proceedings of the Image Intensifier Symposium, Oct, 1958 1958 US Army Engineering Research Development Laboratories, Fort Belvoir, USA.

LI, Z., DAS, A. & ZHOU, J. Theoretical basis for intrusion detection. Systems, Man and Cybernetics (SMC) Information Assurance Workshop, 2005. Proceedings from the Sixth Annual IEEE, 2005. 184-192.

LUKATSKY, A. 2002. *Protect your information with intrusion detection*, A-List.

MUKKAMALA, S. & SUNG, A. H. A comparative study of techniques for intrusion detection. Tools with Artificial Intelligence, 2003. Proceedings. 15th IEEE International Conference on, 2003. 570-577.

POST, G. & KAGAN, A. 1998. The use and effectiveness of anti-virus software. *Computers & Security,* 17, 589-599.

PTACEK, T. H. & NEWSHAM, T. N. 1998. Insertion, evasion, and denial of service: Eluding network intrusion detection. Secure Networks Inc.

ROESCH, M. Snort - Lightweight intrusion detection for networks. Proceedings of USENIX 13th Systems Administration Conference (LISA '99), 1999 Berkeley, CA. 229-238.

SOMMER, P. 1999. Intrusion detection systems as evidence. *Computer Networks-the International Journal of Computer and Telecommunications Networking,* 31, 2477-2487.

About the authors

Chris Tucker is an information security consultant and a Director of Stochastic Systems Ltd. He is a member of the Network Research Group at the University of Plymouth, where he is currently reading for a PhD in intrusion detection systems. He is a Fellow of the British Computer Society and the Institution of Engineering and Technology, holding a BSc in Physics and an MSc in Computing. His research

interests are currently focussed on the application of digital signal processing techniques to information security. He can be contacted at intrusion@stochastic.co.uk.

Steven Furnell is the head of the Network Research Group at the University of Plymouth in the United Kingdom, and an Adjunct Associate Professor with Edith Cowan University in Western Australia. He specialises in computer security and has been actively researching in the area for fourteen years, with current areas of interest including security management, computer crime, user authentication, and security usability. Prof. Furnell is a Fellow and Branch Chair of the British Computer Society (BCS), a Senior Member of the Institute of Electrical and Electronics Engineers (IEEE), and a UK representative in International Federation for Information Processing (IFIP) working groups relating to Information Security Management (of which he is the current chair), Network Security, and Information Security Education. He is the author of over 160 papers in refereed international journals and conference proceedings. He can be contacted at sfurnell@plymouth.ac.uk.

Bogdan Ghita is a senior lecturer and a member of the Network Research Group at University of Plymouth. His main areas of interest are network performance monitoring and modelling, network security, and MoIP communication. As part of research projects he was involved in, Dr. Ghita has published 14 papers in international journals and conference proceedings. He can be contacted at bghita@plymouth.ac.uk. Further details can be found at www.network-research-group.org .

Phil Brooke is a principal lecturer at the University of Teesside, United Kingdom, where he researches security and formal methods. He completed his D.Phil. in Computer Science at the University of York in 1999. Subsequently, he worked as a software engineer in the security domain for two years and then five years as a senior lecturer at the University of Plymouth. He can be contacted at P.J.Brooke@tees.ac.uk.