THE EFFECTIVENESS OF INTRUSION DETECTION SYSTEMS

Charles M. Iheagwara

A Submission

Presented to the University of Glamorgan

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

School of Computing, University of Glamorgan

October 2004

To my parents for giving me life,
and to my wife Susan and daughter Chinenye for sharing it with me.

Acknowledgements

This submission is the result of more than four years of research work with my advisor, Dr. Andrew Blyth and a few others.

I would like to express my gratitude and thanks Dr. Andrew Blyth. He received me with open arms to the School of Computing for my Ph.D. research and provided me with continuous guidance and support throughout the entire duration of the program. I learnt from him many things in the field of Intrusion Detection Systems and Computer Security. He opened my thought process to the best ways to conduct effective research. Dr. Blyth taught me the discipline needed to be a good researcher and kept restoring my hope with these simple words: "excellence, diligence, perseverance, hard work, quality, organization!" I'd like to thank him for sticking out with me all these years and giving liberally of his time, especially in the final home stretch.

My thanks also go to Professor G.E. Taylor, Director of graduate studies. He provided invaluable feedback and insightful advice for this research work.

And I thank the members of the Department Research Committee: Professor Dunkerly, Dr. Iain Sutherland and Mr. David Eyres whose guidance got me back on track and in motion several times when I was fogged in.

An essential contribution to every work is given by discussions and joint works. It is impossible to list all the researchers from whom I received valuable contributions and support through out the period of my PhD studies. Many thanks to Professor Mukesh Singhal of the University of Kentucky who contributed to some of the research work and Kevin Timm and David Kinn who contributed significantly to the research studies on IDS implementation, management and Return on Investment.

Many others also provided support for the research studies. I wish to mention the IT Security group at Edgar-online, Inc. In particular, I thank Richard Jones, Javier Vacaflores and Georges Burgos for their useful suggestions. Many thanks to Ramiz Mansoor of Internet Security Systems who provided me with valuable information on the deployment and operations of enterprise-wide IDSes and as well, assisted with the experimental setup of the research work.

I want to also thank Kerry (Kwasi) Holman, Wanda Plumer, Necola Shaw, Roxane Grissett, Susan Ejimofor, Rama Parameswaran and the entire staff of the Prince Georges County Economic Development Corporation, Maryland, USA for their encouragement and moral support.

I am indebted to Alun Evans for giving his time (whether by phone, or by e-mail), as we went through sorting out the administrative details of this programme. My special thanks goes to Kim Merritt of the Finance Department who brightened my experience with her

humor and friendship.

Of course, the works presented in this submission is strongly based on what preceded the last four years. For all kinds of support, courtesy, love, patience and encouragement, I thank my wife Susan, my daughter Chinenye, my mother Lady Bernadette Iheagwara, and the other members of my extended family for standing with me through out the period of this program.

And I am grateful to God the almighty father, for his grace, blessings and protection.

Finally, I would like to thank in a very big way someone who is not here anymore and who would have been happy to know about me and the progress of my Ph.D. work. Sir Livinus Iheagwara was a great father who taught me the value of hard work and the rewards of patience. He taught me to love and serve the Lord Jesus Christ and to love my fellow men and women. To a great father, I dedicate this work.

TABLE OF CONTENTS

Abstract

Charles Iheagwara, Ph.D., University of Glamorgan, October 2004. The Effectiveness of Intrusion Detection Systems. Advisor: Dr. Andrew Blyth.

This study investigates the following hypothesis: "The effectiveness of intrusion detection systems can be improved by rethinking the way the IDS is managed and by adopting effective and systematic implementation approaches. "

This submission introduces the work done to show the validity of this hypothesis. It demonstrates its practicability and discusses how different technical factors; local environmental (systems/network) factors; implementation and management factors affect intrusion detection systems effectiveness.

We conduct studies on intrusion detection systems to expand our knowledge of their basic concepts, designs, approaches and implementation pitfalls. We analyze implementations of the major intrusion detection systems approaches/products and their inherent limitations in different environments.

We discuss the issues that affect intrusion detection systems effectiveness and explore the dependencies on several components, each of which is different and variable in nature. Then, we investigate each component as a separate and independent subhypothesis.

To provide evidence in support of the hypothesis, we conduct several studies using different approaches: experimental investigations, case studies, and analytical studies (with empirically derived arguments).

We develop methodologies for testing intrusion detection systems in switched and gigabit environments and perform tests to measure their effectiveness against a wide range of tunable parameters and environmentally desirable characteristics for a broad range of known intrusions. The experimental results establish the impact of deployment techniques on intrusion detection systems effectiveness. The results also establish empirical bandwidth limits for selecting appropriate intrusion detection technologies/products for highly scalable environments.

Through case studies, we demonstrate how management and implementation methods affect intrusion detection systems effectiveness and the Return on Investment.

Finally, in our analytical work we illustrate how systems configuration settings and local security policies affect intrusion detection systems effectiveness.

Together, the results provide the evidence in support of the hypothesis and, hence, we contribute to the existing body of knowledge by suggesting and demonstrating the ways to improve the effectiveness of intrusion detection systems.

1 Introduction

The effectiveness of intrusion detection systems (IDS) is dependent on many factors including an organization's implementation strategy; and how well the management of the IDS technology helps the organization achieve the tactical and strategic objectives it has established.

This assertion is the thrust of this research and in this submission overview, I will summarize the work done to show its validity.

Prior to reviewing the studies, the overview presents background information on intrusion detection systems (IDSes) in Section 1.1 and review of the problems with IDS implementations in Section 1.2. The "hypothesis" will be presented Section 1.3 and the research approach in Section 1.4. The organization of the document will be outlined in Section 1.5.

1.1 Background

The following background information will help the reader understand the mission, historical evolution, techniques, architecture and implementation of IDSes.

1.1.1 The IDS Mission

Intrusion detection system is a security technology that attempts to identify and isolate ``intrusions'' against computer systems, i.e. the IDS monitors computer systems and network traffic and analyzes that traffic for possible hostile attacks originating from outside the organization and also for system misuse or attacks originating from inside the organization. Given the above, the main task of the IDS is to defend the computer system by detecting and possibly repelling attacks to it.

Intrusion detection systems evolved due to the lack of intrusion prevention systems (IPS: defined as in-line products or systems that focuses on identifying and blocking malicious network activity in real time) and the following issues:

- It is impossible to build a completely secure system in today's software development environment because the programming languages and operating systems used for development and implementation introduce a number of security flaws [1]. These security flaws are difficult to detect and intruders can use these flaws to bypass existing security mechanisms.

- There is usually a transition period measured in decades (in terms of security)

during the replacement of a large number of operating systems and applications with more secure ones.

- Existing cryptographic systems are not completely secure and have exploitable weaknesses for a determined and resourceful intruder. The best cryptographic system offers no protection against lost or stolen keys or poorly chosen passwords.

- There is an inverse relationship between the level of system security and user efficiency. As system security increases, user efficiency decreases. A completely secure system, with existing security techniques, is practically unusable.

- Finally, a secure system may still be vulnerable to an insider misusing their privileges.

1.1.2 Historical Evolution

Historically, the evolution of IDSes followed a systematic and sequential order of events. The concept of IDS evolved in 1980, when James Anderson first proposed that audit trails should be used to monitor threats [2]. Prior to this, the importance of audit trails on data was not evident as all the available system security procedures were focused on denying access to sensitive data from an unauthorized source. Following Anderson's suggestion was a proposal in 1987 by Dorothy Denning on the development of an Intrusion Detection System abstract model [3]. In effect, the abstract model was the first to propose the concept of intrusion detection as a solution to the problem of providing a sense of security in computer systems. Industry watchers saw the model as more of a retrofit approach, in comparison to the traditional proactive methods of encryption and access control.

Following Denning's proposal were a series of efforts to come up with an enhanced model and prototypes. Teresa Lunt et al. in 1988 refined Denning's model by creating IDES (Intrusion Detection Expert System) [4] designed to detect intrusion attempts against a single host. In 1995 an improved version called NIDES (Next-generation Intrusion Detection Expert System) [5]was developed. Other systems include the Haystack system [6] developed in 1988 to assist Air Force Security Officers detect misuse of the mainframes used at Air Force Bases, and in 1989 MIDAS (Multics Intrusion Detection and Alerting System) [7] developed for the same reasons, but for the National Computer Security Center's Multics mainframe. Wisdom and Sense [8] was developed in 1989 from the Los Alamos National Laboratory, and Information Security Officer's Assistant (ISOA) [9] from Planning Research Corporation.

The nineties saw a phenomenal increase in the scope and breadth of research and development of IDS technologies. Among these was the introduction in 1990 of a new concept - Network Security Monitor (NSM), now called Network Intrusion Detector or NID [10]. NID examines suspicious behavior by passively monitoring the network traffic

in a LAN. In the following year (1991) instead of examining the audit trails of a host computer system, a different idea was introduced with NADIR (Network Anomaly Detection and Intrusion Reporter) [11] and DIDS (Distributed Intrusion Detection System) [12]: the audit data from multiple hosts were collected and aggregated in order to detect coordinated attacks against a set of hosts. Mark Crosbie and Gene Spafford [13] in 1994 suggested the use of autonomous agents in order to improve the scalability, maintainability, efficiency and fault tolerance of IDS. With the design and implementation of GrIDS [14] in 1996, a new approach to address the scalability deficiencies in most contemporary intrusion detection systems was introduced. GrIDS facilitates the detection of large-scale automated or coordinated attacks, which may even span multiple administrative domains. Ross Anderson and Abida Khattak in 1998 offered an innovative approach to intrusion detection, by incorporating informational retrieval [15] techniques into intrusion detection tools.

1.1.3 Techniques

The techniques for intrusion detection can be divided into two main types.

Anomaly Detection: In anomaly detection techniques it is assumed that all intrusive activities are necessarily anomalous. This means that by establishing a "normal activity profile" for a system, it is possible, in theory, to flag all system states varying from the established profile by statistically significant amounts as intrusion attempts.

The fundamental issues in anomaly detection systems is the selection of threshold levels so that anomalous activities that are not intrusive are flagged as intrusive and intrusive activities that are not anomalous result in false negatives (i.e. events are not flagged intrusive, when they actually are). Anomaly detection systems are also computationally expensive because of the overhead of keeping track of, and possibly updating several system profile metrics. The block diagram of a typical anomaly detection system is shown in Figure 1.

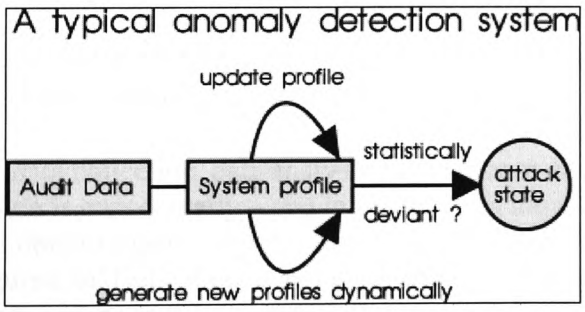


Figure I: A typical anomaly detection system [16].

Misuse Detection: The idea behind misuse detection schemes is that we can represent

attacks in the form of a pattern or a signature such that any variation of the same attack can be detected. This means that misuse detection systems try to recognize known "bad" behavior or attack patterns.

The difficulties in misuse detection systems include discerning how to write signatures that encompass all possible variations of the pertinent attack and signatures that do not also match non-intrusive activity. The block diagram of a typical misuse detection system is shown in Figure 2 below.

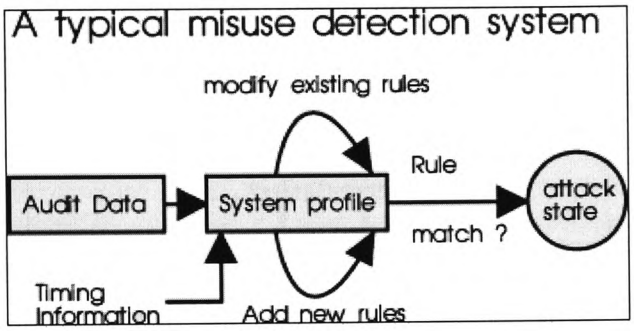


Figure 2: A typical misuse detection system [16].

1.1.4 Architecture

The current architecture (Fig.3) of commercially available IDS products is built primarily out of the perceived role/tasks of the IDS in information systems assurance.

By design, an IDS is made up of three components:

- Information sources,
- Analysis, and
- Response.

The three components are seamlessly integrated and are structured in a sequential order to maintain the functionality of the system.

Three distinct phases: data collection, data analysis and response characterize the system. Thus, intrusion detection is conceptually – and in practice - in most cases accomplished as follows: the system obtains event information from one or more information sources, performs a pre-configured analysis of the event data, and then generates specified responses, ranging from reports to active intervention when it detects intrusions. There is also a management system that allows a security or network administrator to monitor and configure the system and to analyze the data. These components may or may not be running on the same box, and all of them may not be present.

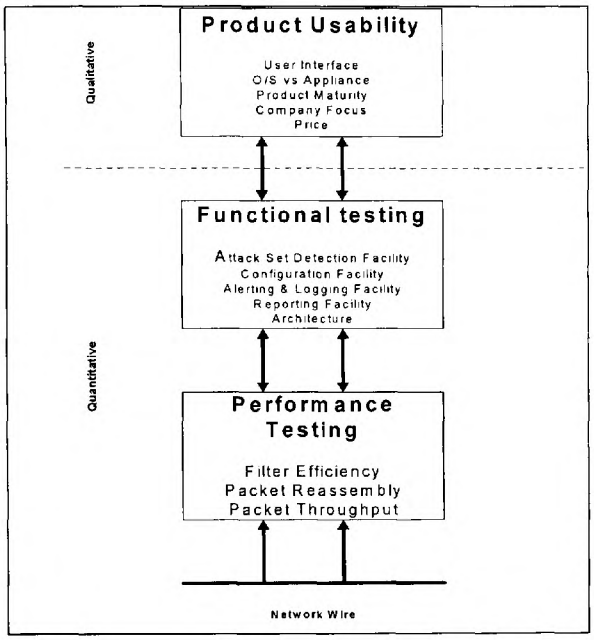


Figure 3. The standard IDS architecture [17].

Thus, the IDS is an assembly of different components all of which functionally relate to each other singularly or wholly.

At the component level (Fig.4) the IDS always has its core element - a sensor (an analysis engine) - that is responsible for detecting intrusions. This sensor contains decision-making mechanisms regarding intrusions. Sensors receive raw data from three major information sources: own IDS knowledge base, syslog and audit trails. The syslog may include, for example, configuration of file system, user authorizations etc. This information creates the basis for a further decision-making process.

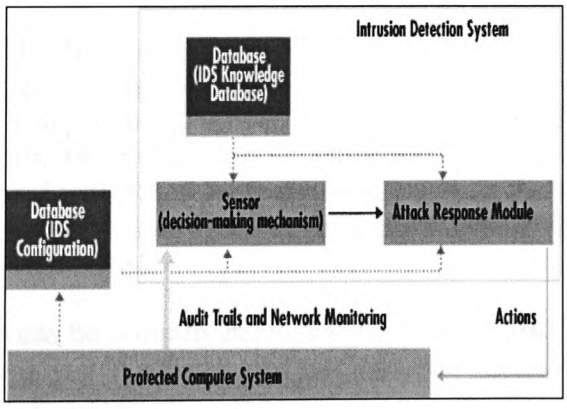


Figure 4: A sample IDS [18].

The component/mechanism responsible for data/information collection is integrated with the sensor (Fig.5) [18] — an event generator. The method of collection is determined by the event generator (which is usually the operating system, network or application) policy that defines the filtering mode of event notification information and produces a policy-consistent set of events that may be a log (or audit) of system events, or network packets. Storage of the policy set can either be in the protected system or outside.

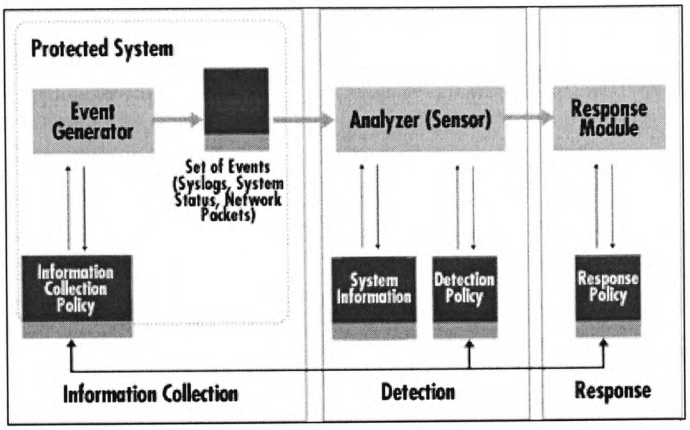


Figure 5: IDS components [18].

As for detection, a detection policy database (an information repository) is used for analysis by the analyzer. A typical content includes the following: attack signatures, normal behavior profiles, and necessary parameters for example, thresholds. Additionally, the database holds IDS configuration parameters, including modes of communication with the response module. Also, the sensor also has its own database containing the dynamic history of potential complex intrusions.

1.1.5 Implementation

The implementation of the IDS can be accomplished in two ways (Network and Host-based) depending on specific needs and local system environmental conditions. Thus, the IDS is an overlay of two separate and different Network-based (NIDS) and Host-based IDS (HIDS) technologies. This means that the whole network or any subsets of the network can be monitored with NIDS from one location while the HIDS can be deployed to watch specific hosts critical servers such as databases, Web services and essential file servers to significantly reduce risk.

Topologically, the IDS can be centrally deployed (for example, physically integrated within a firewall) or deployed in a distributed fashion (a distributed IDS consists of multiple Intrusion Detection Systems (IDS) over a large network, all of which communicate with each other). For more sophisticated systems, the agent structure

principle (where small autonomous modules are organized on a per-host basis across the protected network [19]) is employed. In this case, the role of the agent is to monitor and filter all activities within the protected area and — depending on the approach adopted — make an initial analysis and even undertake a response action. Also, IDS can employ more sophisticated analysis tools to aid with the detection of decisive distributed attacks [20] with the agent roaming across multiple physical locations. Thus, agent type factors into the implementation scheme when introducing new policies in response to new types of attacks [21] and IDS agent-based solutions also use less sophisticated mechanisms for response policy updating [22].

1.2 Problems with Existing Intrusion Detection Systems

This brief review serves to inform the reader of the magnitude, scope and nature of the problems that diminish the IDS effectiveness.

There are several unique obstacles that limit the performance effectiveness of commercially implemented IDS products. Primarily these are:

1. Issues with variant signatures,
2. Excessive number of (false positives and negatives) alerts,
3. Data overload,
4. Scalability issues,
5. Issues in Large-scale deployment,
6. Difficulties in switched environments,
7. Cost-effectiveness issues, and
8. Management Issues.

1. Issues with variant signatures: While the ability to develop and use signatures to detect attacks is a useful and viable approach, there are shortfalls to only using this approach that should be addressed. The problem is that signatures are developed in response to new vulnerabilities or exploits that have been posted or released. Therefore between the creation of an attack and the deployment of a signature, a window of opportunity exists for an intruder to mount an attack with little to no chance of the attack (***zero day attack***) being detected. *If the attack takes place, the IDS will be ineffective in detecting it, hence its primary mission is defeated.*

2. Excessive number of alerts: A common complaint in IDS deployment is the amount of false positives the IDS generates. Developing unique signatures is a difficult task and often times the vendors will err on the side of alerting too often rather than not enough. It is much more difficult to pick out a valid intrusion attempt if a signature also alerts regularly on valid network activity. A difficult problem that arises from this is how much information can be filtered out without potentially missing an attack. *Overall, this problem results into a management overburden, wastage of network resources and higher operating cost.*

As for false negatives, the issue is not detecting attacks for which there are no known signatures when they occur (*e.g. zero-day exploits).* The result is that the IDS do not generate an alert when an intrusion is actually taking place. *Hence it is ineffective.*

3. Data overload: Another important factor, which could diminish the effectiveness of IDS performance, is the volume of data to be analyzed. In effect, how much data an analyst can effectively analyze becomes very important. With generation of excessive alerts and the large volume of transactions to be analyzed, the amount of data he/she needs to look at becomes so large and definitely an overburden. Depending on the

intrusion detection tools employed and its size, there is a possibility for logs to reach millions of records per day. *This problem results into a management overburden and makes the implementation costly (expensive) due to the extra financial expense incurred.*

***4. Scalability Issues**:* In the last couple of years, there has been a significant increase in network traffic. As a result, the Gigabit Ethernet technology was introduced to accommodate this increase in bandwidth – and thus the volume of traffic to be analyzed. The problem associated with this is that with Gigabit traffic, the older IDS technologies that operate at 10mbps or 100mbps bandwidths become seriously overloaded. *And after a certain point, the performance takes a nosedive to the point of the IDS being completely ineffectiveness*.

5. Issues in Large-scale deployment: Significant differences exist between implementation of the IDS in small and large enterprise systems. The most obvious difference is that in large enterprise implementations there are more endpoint machines (computers, servers, and network segments) that must be protected. This will lead to longer installation time and a more complicated set up in terms of systems configuration and optimal selection of sensor placement within the network. Also smaller enterprises, by definition, have less choices and options about where to strategically install the IDS. By contrast, larger enterprises must often spend days or even weeks deciding on the optimal placement of IDS agents, managers, and IDS configuration groupings. *This problem results into improper technical deployment and mismanagement of the implementation which diminishes the IDS effectiveness.*

6. Difficulties in switched environments. Switched and/or high-speed networks create problems for IDSes: many are unreliable at high speeds, dropping a high percentage of network packets; and switched networks often prevent IDS network interface cards (NICs) that operate in promiscuous modes from seeing passing packets. This problem is compounded in large-scale deployments where multiple layers of switches inundate the network. *The overall concern is the scope and visibility of the IDS which directly affect its performance effectiveness*.

7. Cost-effectiveness: Given the high cost of IDS deployments especially when multiple deployments are involved, organizations must justify implementation expenses by proving that the IDS is cost-effective. One possible justification is to establish that the deployment of the IDS should lead to a reduction in the annual loss expectancy (ALE) and the return on security investment (ROSI).

But as a result of several factors, IDS implementations have not always been cost-effective. Improper methods have been used to purchase, deploy and manage the devices in some organizations. *This results into a more expensive implementation.*

8. Management Issues: The manner in which the IDS is implemented will affect its

effectiveness. In some organizations, the IDS is implemented without due consideration to proper management approach. Also certain implementation management decisions are made on ad hoc basis. Typical issues here include inadequate manpower, improper selection of implementation technique, lack of training, etc. *This often results into poor implementations and poor IDS performance*.

Together, all of the above problems diminish the ability of the IDS to function effectively and possibly resulting in deployments that are unprofitable. Hence there is the need to rethink the whole way the IDS is implemented and managed.

1. 3 Hypothesis

The effectiveness of intrusion detection systems (IDS) can be improved by rethinking the way the IDS is managed and by adopting effective and systematic implementation approaches.

Subhypotheses

Six distinct subhypotheses are implicit and not exclusive in the *above* hypothesis, each of them proposing a different or varying aspect (component) of IDS effectiveness:

1. Deployment techniques affect the IDS effectiveness.
2. The product/technology used to implement the IDS in different environments affect the IDS effectiveness.
3. The manner in which the IDS is managed affects its performance effectiveness and Return On Investment.
4. Cost-effective implementation approaches will lead to a positive Return On Investment.
5. System configurations settings play a role in IDS effectiveness.
6. Tailoring the IDS function to be more consistent with local security policy improves the IDS effectiveness.

These subhypotheses were examined for their independent merits, and each one was tested separately and regarded in the context of showing validity of the primary hypothesis. Elimination or confirmation of one subhypothesis does not imply elimination or confirmation of the others. And the individual weights of the subhypothesis may be different depending on implementation.

1. 4 Approach to Research

The effectiveness of intrusion detection systems depends on so many issues/aspects. For the purpose of the research studies, the issues investigated were classified into four (4) distinctive factors (Figure 6).

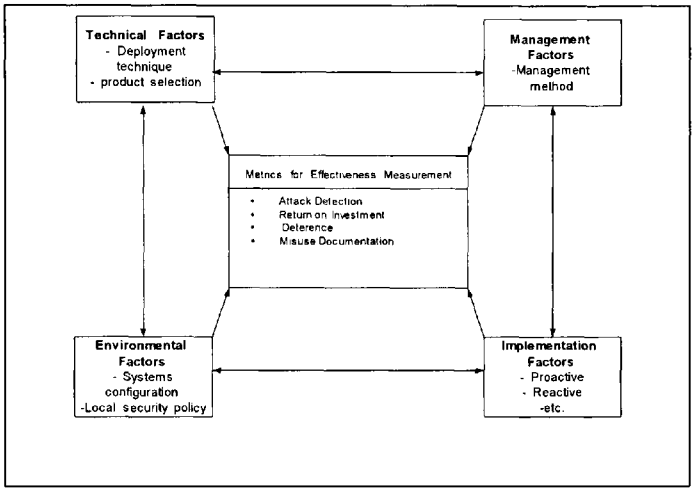


Figure 6: IDS Effectiveness Determining Factors.

As can be seen in Figure 6, each factor consists of selected research issues that are similar in nature yet distinctive and at times relate to one another in a collaborative manner. As a result, some investigations cover a period spanning many years while the others were investigated in a relatively shorter time frame. Consequently, this review will not follow a strict chronological order in terms of the date(s) of investigation.

The approach to this research is shown in Figure 7. Essentially, the approach is to prove the thesis hypothesis through different study approaches: experiments, case studies, and analytical and empirically derived arguments.

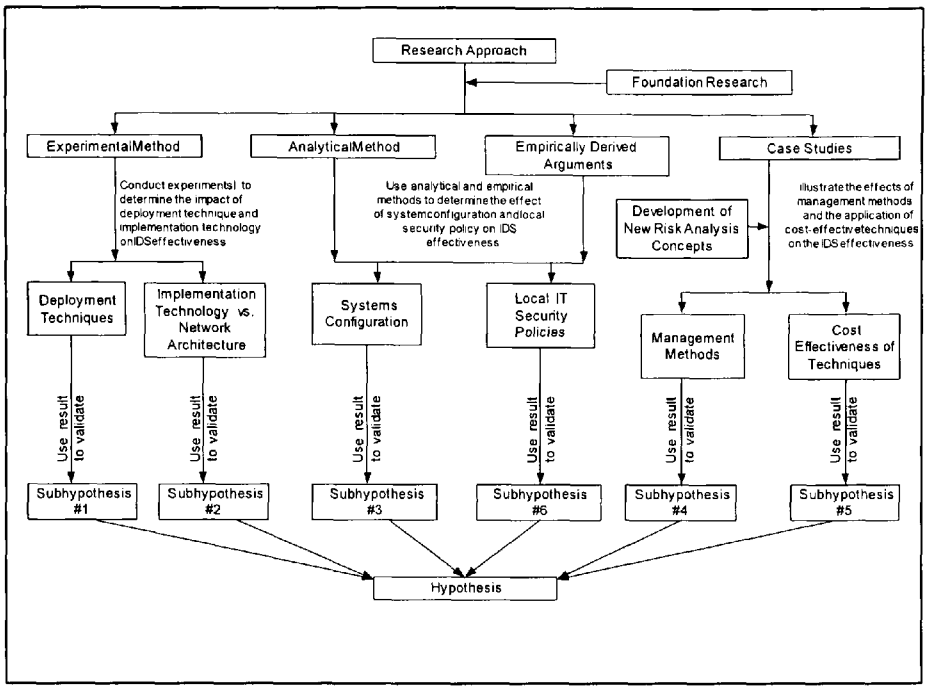


Figure 7: IDS effectiveness research approach.

1.4.1 Foundation Research

Because an investigation of the factors affecting IDS effectiveness requires a thorough knowledge of IDS design, function and implementation approach, we commenced our research with the exploration of IDS concepts and approaches from the perspective of the engineering of IDS design; implementation and operational environments. This exploration set the stage and provided the context upon which we investigated, discussed and derived solutions pertinent to IDS design, operational capabilities, management and implementation in enterprise systems.

In the studies, we illustrated the challenges and benefits of designing flexible IDS types, rethinking IT security policy pertinent to the design and management of IDS products, and adjusting decision-making processes to depend on adaptive technical information.

Drawing on several organizational and technological perspectives, we examined the design and dynamics of growth of IDS implementations and evaluated the concepts and approaches of the IDS and performed a comparative analysis of each IDS types; and the suitability of their use for certain environments.

We also sought to understand the relationship between IT security policies and the functionality of IDS products, and explored the use of formal methods to specify, verify

and validate secure IDS system properties

Contribution: These studies provided the contexts and analytical frameworks used to conduct studies on IDS effectiveness.

These studies are reviewed in Chapter 2.

1.4.2 Experimental Research

The next stage of the research involved experimentally investigating the relationships between underlying technical and implementation factors and the IDS effectiveness. On this, two performance evaluation experiments were conducted to:

(i) Establish the effects of different deployment techniques on IDS performance effectiveness; and
(ii) Establish the relationship between using different IDS types in different networking technologies (Mbps vs. Gbps) and their individual and or collaborative effects on the IDS effectiveness.

These experiments were remarkable for two reasons:
(i) The test beds for the experiments were switched production networks, which is a marked departure from earlier simulation-based research studies.
(ii) The second experiment was conducted on a production network with gigabit throughput that is increasingly becoming the norm in networking architecture. The impact here is that the empirical values established in the studies, could on specific basis serve as benchmarks especially for network architectures similar to those described in the test bed.

Contribution: From these studies:

- ***We developed methodologies for testing IDSes in switched environments and; to evaluate the performance of different IDS products with different design architectures in different environments (Mbps vs. Gbps).***

- ***We established the best techniques to deploy the IDS within a switched network environment. This could be used for organizations trying to justify and/or optimize IDS deployment techniques so as to maximize performance effectiveness.***

- ***Finally, we established empirical bandwidth values and limits upon which selection decisions can be based on the use of multiple 100Mbps IDS sensors instead of a single Gigabit IDS sensor in environments with bandwidths exceeding 100Mbps to enhance the performance effectiveness.***

These studies are reviewed in Chapter 3.

1.4.3 Case Studies

Due to the need to establish the relationship between implementation and management methods and IDS effectiveness, we used case studies to investigate the impact of implementation and management factors on IDS effectiveness.

In the studies we quantified the IDS effectiveness in risk analysis and financial terms (Annual Loss Expectance, Return on Investment). Overall, three studies were conducted:

(i) The tasks for the first study were to derive new formulas and expanding our understanding of risk concepts that takes into account the nuances associated with IT security environments. This is in realization of the fact that modification of risk analysis concepts and the formulas is decisive in establishing reliable quantifiable measures (i.e. ROI, ALE) with which to gauge the IDS performance effectiveness. This study primarily provided the computational formulas and analytical frameworks to (ii) and (iii) below.

(ii) The second study investigated the effects of management methods on IDS effectiveness. The challenge is to demonstrate that IDS effectiveness is dependent upon an organization's deployment strategy and how well the management of the technology helps the organization to achieve the tactical and strategic objectives it has established. The ultimate goal is to prove the value proposition (re: a benefit in the form of a quantifiable reduction in ALE) i.e. the method in which IDS devices are managed can have a serious effect on the ALE and the ROI.

(iii) In the third study, we addressed the problem of bridging the gap between the technical solutions that the IDS provide and the business need for it. The challenge is to formulate cost-effective management frameworks that can be used to adjust the usage of different IDS implementation techniques.

Contribution: The following are the results and contributions of the case studies:

- ***We introduced and demonstrated the application of a new concept – "the Critical Threat Multiplier (CTM)".***

  The idea behind the CTM is that a security compromise incurs two types of costs:

  a) The direct cost of lost integrity/confidentiality/availability, and

  b) the indirect cost, of the compromised component serving as a potential Stepping-stone for future attacks.

  The CTM tries to capture the second type of costs, which are typically ignored in the classic risk analysis framework.

- *We developed a model that can be used to determine the viability of different management approaches and how they affect the IDS ROI; and how to accurately calculate the ROI for IDS implementations using the CTM concept.*

- *By applying financial risk calculations to demonstrate the value of deploying IDS with different supporting procedures, we opened a new way of measuring IDS effectiveness.*

- *We conceived of strategies and approaches to support effective decision-making about which techniques are appropriate for the cost effective management of the IDS in a given environment.*

- *We also developed a scheme that involves first performing a risk analysis that produces a cost matrix for the assets under attack, and then independently calculating damage, response, and operation costs for those assets. Then, we developed the frameworks that can be used to analyze site-specific cost factors for IDS implementation.*

These studies are reviewed in Chapter 4.

1.4.4 Analytical Studies

The study sets out to provide evidence that IDS effectiveness can be impacted by local environmental conditions: network architecture, traffic characteristics, system configuration settings, local IT security policy, etc.

Therefore, in the study, we investigated the impact of local systems configuration and local policies on the IDS effectiveness in order to establish strategies that tailor the IDS function to be more consistent with the local security policy.

Contribution: From the studies:

- *We proposed methods that take a security policy as the basis for the configuration of the IDS components.*

- *We conceived of and proposed several effective techniques for optimizing system configurations so as to make the IDS more responsive and effective to the settings.*

This study is reviewed in Chapter 5.

1.4.5 Research and development studies

To conclude, from the different studies we culled from the likely scenarios sketched in various parts of the research studies the list of some of the most pressing issues and subject matters to be pursued in future research. The details of these are reviewed in Chapters 6.

1.5 Organization of Document

This submission overview is organized as follows: Chapter 1 introduces the reader to background information on the IDS, implementation problems, thesis statement and approach to research. Chapter 2 describes the foundation studies on IDS concepts and approaches. Chapter presents the experimental and analytical studies and in Chapter 4 is a review of the case studies. Chapter 5 describes the analytical work and Chapter 6 discusses future research and development work. Finally, Chapter 7 presents the conclusions and summarizes the contributions of the research studies.

Chapter 2. Foundation Studies

The following studies are reviewed in this Chapter:

1. A Comparative Analysis of Intrusion Detection Systems. [Appendix 1]

2. Security problems and the interaction of security policies in the design and implementation of IDS in enterprise networks. [Appendix 2]

3. The impact of security layering on end-to-end latency and system performance in switched and distributed e-business environments. [Appendix 3]

2.1 A Comparative Analysis of Intrusion Detection Systems

There are a number of classification techniques [23-24] that can be used within intrusion detection approaches. These techniques classify events as either intrusive or normal. They techniques include statistical analysis, predictive patterns, state transition, expert systems, neural networks, machine learning, pattern matching, graph-based and model-based approaches.

Based on these techniques, several IDS approaches [25-36] have emerged over the years. In this study, we provided a systems-based description of intrusion detection technologies and concept-based analytical comparison of the leading implementation approaches and techniques. We also summarized the advantages and disadvantages of each intrusion detection approach and then analyzed the suitability of use of each approach for different environments.

2.2 Security problems and the interaction of security policies in the design and implementation of IDS in enterprise networks

Practical experiences in the implementation of the IDS presents a picture that vividly depict the gaps in the application of sound engineering principles to IDS designs and implementation. This has created a situation where the IDS operate on somewhat different policy settings from the local security policy settings.

Research work abounds on the use of formal methods for the analysis, design and verification of security systems and products but none is evident for IDS product development. The concern here is the correctness of the design and conformance to established security policies.

To make the argument about using formal methods to make the IDS function within verifiable security context, we used inferences and analogies to buttress our points. For instance, a fair analogy is the verification process of general-purpose computer programs, where reliable testing techniques allow many bugs to be detected, but will not provide a basis for complete proof of correctness. In this case, specific methods and implement tools have been designed, in order to aid the initial correct design of cryptographic protocols. This has been achieved by incorporating formal methods into the design process.

Also, transport protocols have been verified and validated using formal methods. For transport protocols, Meadows [36] proposed a stepwise-layered methodology that can be integrated with the Heintze and Tygar's approach [37], which is based on a stack of models at different levels of abstraction. As a first step, the protocol designer uses a relatively abstract model to construct and verify the security protocol. If this protocol is correct at that top layer, the designer focuses on a more detailed model, which refines the abstract one. The repeated execution of this process leads to the final production of a detailed specification. Much of the existing work on requirements specifications has this specific flavor.

Based on the proposition by Meadows [36], Rudolph introduced an approach for designing an abstract model for cryptographic protocols that can be used as the top layer of a layered design method [38]. The main idea is the usage of Asynchronous Product Automata. The whole design process starts with a relatively abstract model at the top layer and ends in a refined specification that can be proven to be an implementation of the top level. This model reaches a higher level of abstraction than the model presented in the work of Heintze and Tygar [37] through the use of logical secure channels, instead of encryption.

Buttyan [39] utilized the notion of channels to present a simple logic for authentication protocol design. These channels are abstract views of various types of secure communication links between principals. The way channels are used is similar to the use

of Pi calculus channel primitives. The proposed Simple Logic preserves the simplicity of the BAN logic and adopts some concepts from the GNY logic. It consists of a language and a small number of inference rules. The language is used to describe assumptions, events, and the protocol goals. The inference rules are used to derive new statements about the system. The goal of the analysis is to construct a witnessing deduction, which is a derivation of the goals from the assumptions and the formal protocol description. The protocol is correct in the case where such a deduction exists. The lack of a witnessing deduction means that the protocol may not be correct.

Gollmann [40] suggested that the design of authentication protocols has proven to be error prone partly due to a language problem. The objectives of entity authentication are usually given in terms of human encounters while we actually implement message-passing protocols. The author proposed various translations of the high-level objectives into a language appropriate for communication protocols.

Several researchers believe that in the near future, more effort will be spent on designing secure protocols and less on formal verifications. Specifically, Meadows argues [36] that design specifications do not guarantee that protocols will meet security goals that were not foreseen by the design approach, that the protocols designed are sometimes impractical, and that - due to the imprecision of design principles - flawed protocols may in any case be designed.

Using the principles enunciated above [36 –41], we visualized instances where the IDS could be designed based on making its functions configurable and interoperable with security policy specifications. And future development of IDS products will be more effective with IDS developers learning from the concepts proposed in formal protocol verification techniques.

Finally, we proposed a conceptual designing testing approach that integrates IT security validation techniques. This approach is developed from basic security properties that can be expected to hold for a variety of design elements. Security policies can be developed abstractly and any particular type of IDS that possesses the required property can then be used in a concrete implementation.

2.3 The impact of security layering on end-to-end latency and system performance in switched and distributed e-business environments.

The imposition of stringent security regimes in contemporary e-business networks to provide a reasonable measure of security for their information systems comes with certain collaterals, some of which are undesirable. The implementation of these security regimes entails formation of a layered architecture (concentric security layers) using packet and application-level filters neither of which provides complimentary functions. The layered architecture provides convenient abstractions and increases the end-to-end latency that results into sub-optimal system performance. IDSes as part of the multi-layer security scheme contribute to the sub-optimality.

The problems associated with stringent security layering must be minimized so that the requirements for performance, reliability, speed and operational support of e-business are not sacrificed. In other words, the implementation of the security scheme should not impede vital system performance indexes such as desirable low values for end-end-latency, Web request-response time, network throughput and protection of the privacy of data. Thus, there is the need to maintain a balance between system performance such as process response time and the security requirements established for the system.

Prior to this research, there were no known studies on the impact of IDS security layering on system performance that are reported in scientific literature. Although a few studies [43, 44] explored the effects of multiple disk use, low-bandwidth modem client connections and throughput on the performance of Proxy Servers that are used to implement stringent security for internally protected information systems. The studies found that the latency advantage of caching proxies vanishes in front of modem connections.

Based on the above, we pioneered this study to investigate the contribution of IDSes (when used as part of a security multi-layer) to end-to-end latency and the resulting degree of sub-optimality of system performance in a distributed and switched e-business network.

The test bed for the experiment was a switched and distributed network. The setup and experimental procedure are discussed in Appendix 3.

The results of this research study established empirical values for end-to-end latency; and the resulting degree of sub-optimality of system performance attributable to the deployment of the IDS in an e-business network.

Chapter 3. Experimental Work

The two experimental studies reviewed in this Chapter are:

1. Evaluation of the Performance of ID Systems in a Switched and Distributed Environment: The RealSecure Case Study. [Appendix 4]

2. A Comparative Experimental Evaluation Study of Intrusion Detection System Performance in a Gigabit Environment. [Appendix 5]

3.1 Evaluation of the Performance of Intrusion Detection Systems in a Switched and Distributed Environment: The RealSecure Case Study.

The performance of IDSes has always been a crucial factor for organizations trying to implement intrusion detection technologies due to a number of reasons including the need to deploy the right IDS product so as to enhance the return on investment.

One of the most obvious ways to measure or gauge the performance of the IDS is to quantify attack detection rates of the IDS. This at times is untenable for a number of reasons including the complexities of network architectures in which the IDS operate.

Previous research studies [41, 42] on IDS detection limits and accuracy have been conducted using simulation techniques within a narrow span of systems parameters. Porras and Valdes [41] discussed IDS failures in terms of deficiencies in accuracy and completeness, where accuracy reflects the number of false positives and completeness reflects the number of false negatives. Richards [42] evaluated the functional and performance capabilities of the industries leading commercial IDS products. In the areas tested, the performance of the IDS was rated based on their distinctive features.

None of these or other documented studies was conducted on a switched network environment, which is typical of many of today's network architectures. Hence, our primary task was to extend the studies to actual switched network environments.

In our research, we leveraged the work of Richards [42] to an actual network built on distributed and switched architecture. We explored the relationship between deployment techniques and the performance of the IDS in a distributed and switched network infrastructure using the RealSecure software suite.

We developed a methodology for testing IDSes that addresses these difficulties faced by the IDS in switched environments. The methodology consists of general software-testing techniques, which I have adapted for the specific purpose of testing the IDS. We first identified a set of desirable characteristics for the IDS such as the ability to detect a broad range of known intrusions. Then, we developed strategies for selecting test cases and detailed testing procedures.

Finally, we used the methodology to test the IDS detection rates in different locations in the switched network. Essentially, this approach helped to establish the relationship between the scope of visibility of the IDS at the different locations and the detection rates.

The contribution to the body of knowledge is that we extended previous research works on IDS performance evaluation to a production network with switched architecture and established empirical values for the IDS capability for different visibility scopes. Therefore, the results (Figure 8) provided a view into the IDS performance in switched

production networks of which most contemporary networks are built.

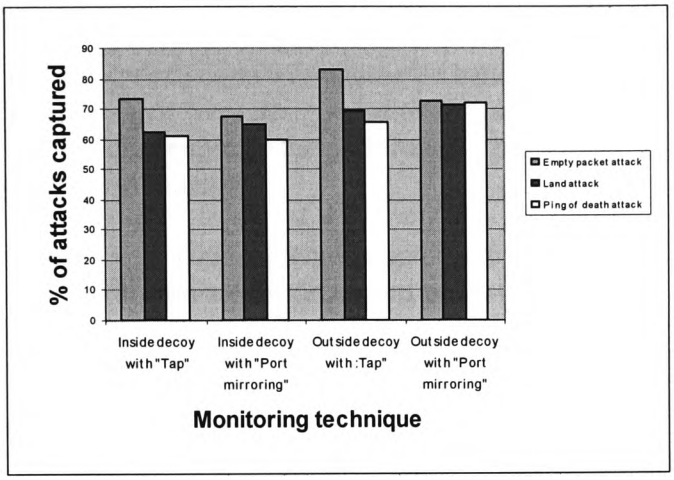


Figure 8: Percentage of attacks captured at 40% network utilization.

In specific terms, we established that the detection capability of the IDS diminishes with increase in bandwidth utilization and as the topology becomes more switched. The implication is that better performance could be achieved with the use of multiple sensors and deployment with packet loss-limiting devices. Also, deployment at network or domain entry points i.e. outside decoy (where the topology is not switched) produces a better performance result. Hence the scope of visibility is an important factor for intrusion detection.

The importance of this study is that we are now able to get a better understanding of the IDS performance in real enterprise system settings. This could be used to provide the justification about which deployment technique will produce a better result and is thus preferable for similar environments.

Subhypothesis Supported

This result supports the assertions of subhypothesis (1)) i.e. "Deployment techniques affect the IDS effectiveness."

3.2 A Comparative Experimental Evaluation Study of Intrusion Detection System Performance in a Gigabit Environment.

The advent of Gigabit network to cope with increased bandwidth demand presents a serious challenge for IDSes that were designed with Megabits bandwidth throttling in mind. At the heart of this challenge are the performance and the efficacy of the 100Mbps IDS sensors in Gigabit environments.

One of the main problems with IDS implementation is the selection of a suitable IDS technology/product for highly scalable environments where Gigabit architectures are used. This problem is particularly acute when trying to select a particular IDS product (from the large number of available IDS devices) for deployment in distributed large networks.

The problem is that currently available commercial IDS products were designed to accommodate traffic with bandwidth not exceeding 100Mbps. Deployment of these products on Gigabit traffic results in poor performances.

Generally speaking, there are options available:

1. Deploy multiple 100 Mbps sensors;
2. Deploy a single Gigabit (Gbps) sensor.

Which of these two might an IT security manager recommend? The answer is not simple. For sure the effectiveness of whichever is selected is the essence. Thus, the challenge is to make a proper choice.

As at the time of this research work, guidance on how to make the selection were not established. Also, the efficacy or the advantage of any of the two options has not been established through an independent investigation/evaluation, although one or two IDS products have been introduced as Gigabit sensors and have been touted to dramatically increase component performance and functional opportunities, possibly leading to dramatically changed system balance and overall performance. But, their operational performance has not been established.

Against this background, we conducted an experimental research that examined the system benefits of using a single Gigabit IDS sensor instead of multiple Megabit sensors in a Gigabit traffic stream for a wide range of defined system attacks, network traffic characteristics, and contextual operational elements.

In the experiment, we first developed a probabilistic methodology to be used to determine the performance of the IDS in a Gigabit traffic stream. Then, employing the misuse attack detection technique we tested the ability of the IDS to detect attacks under varying test parameters. Finally, we analyzed the experimental results, quantified the IDS performance and compared the different values in the context of network

architecture/traffic volume: (Gigabit vs. Megabit), and deployment type: multiple mbps sensors vs. a single gbps sensor.

The experimental results (Figures 9 and 10) established empirical bandwidth limits for IDS effective performance in Gigabit environments. This is a differential marker (benchmark) that can be used to determine when multiple 100Mbps IDS sensors can be used preferentially against a single Gigabit IDS sensor in bandwidths exceeding 100Mbps.

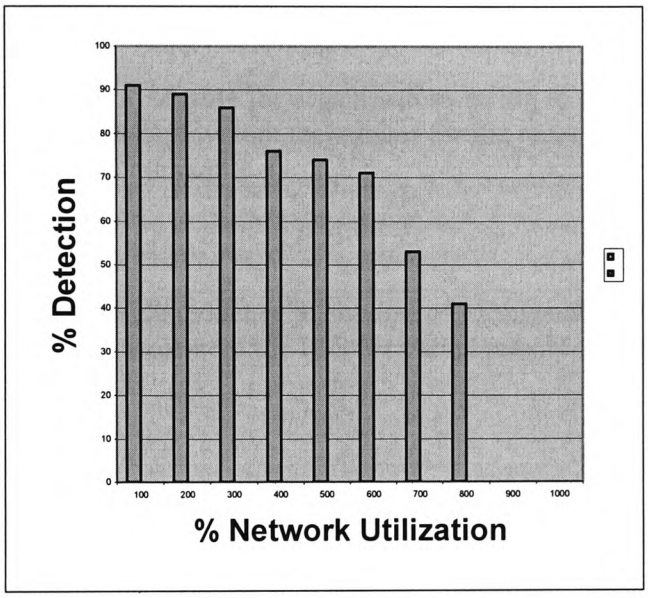


Figure 9: Probability of detection vs. % Utilization with NetworkICE (Gigabit) sensor.

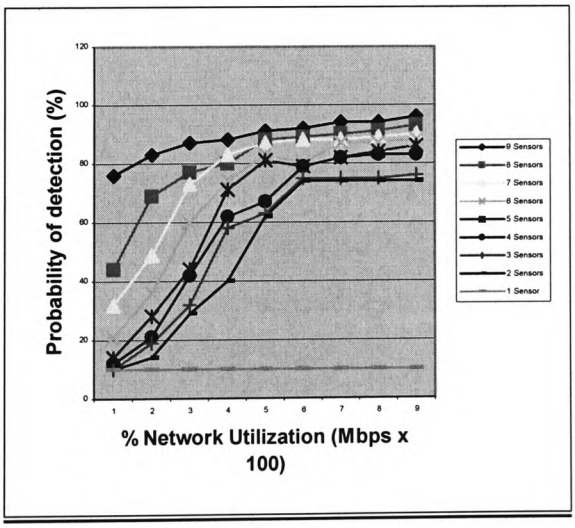


Figure 10: Probability of detection vs. % utilization with Multiple RealSecure (Megabit) sensors.

The contribution of this study to the accumulated body of knowledge is two-fold:

(1) Firstly, we developed a probabilistic methodology that can be used to evaluate and compare different implementation with different IDSes (products and types) for different environments (Mbps vs. Gbps),

(2) Secondly, we established the performance limits and suitability of use of the major IDS approaches (Megabit vs. Gigabit) in high traffic volume environment.

This will be invaluable for organizations trying to optimize IDS product selection with deployment techniques for the most performance effectiveness attainable.

Subhypothesis Supported

The results of this study provide evidence in support of subhypothesis (2) i.e. "The product/technology used to implement the IDS for varied conditions of network traffic affect the IDS effectiveness."

Chapter 4. Case Studies

The following studies are reviewed in this Chapter:

1. Towards an Effective Risk Assessment Methodology: Factoring in Novel Concepts For Assessing Intrusion Detection Systems in Complex Infrastructures. [Appendix 6]

2. The Impact of IDS Deployment Technique on Threat Mitigation [Appendix 7]

3. The Effect of Intrusion Detection Management Methods on The Return on Investment. [Appendix 8]

4. Cost Effective Management Frameworks for Intrusion Detection Systems. [Appendix 9]

4.1 Towards an Effective Risk Assessment Methodology: Factoring in Novel Concepts for Assessing Intrusion Detection Systems in Complex Infrastructures

Determining the value of the ALE for security products in complex environments using conventional cost/benefit (risk) assessment method is quite complex due to the difficulty of coming up with accurate asset values or replacement costs within the organization — variables that are critical to a risk analysis. Asset values are factored into the calculations for Single Loss Expectancy and Annual Loss Expectancy. If accurate asset replacement values cannot be obtained then the risk analysis will yield incorrect results. Further, the determination of the asset value when there is interdependence in networked environments could be extremely difficult because the asset value must be taken in up and down stream dimensions. And for the IDS, measuring the asset value in so many dimensions and in tangible and intangible measures can be challenging.

Devising effective risk analysis techniques for the IDS in complex environments requires re-examination of the basic concepts, assessment approaches, and risk analysis formulas.

Until recently, risk assessment of IDS products has received little or no attention from the information security community for various reasons including lack of statistical data for the asset valuation of IDS products in networked environments and lack of awareness. Also the benefits of organizations implementing IDSes have been seen mostly from technical and not risk management or financial perspective and available risk assessment techniques [45-46] were developed for other purposes as they do not take into account all the "tangibles" and "intangibles" necessary to accurately conduct risk assessments for networked security products like Intrusion Detection Systems (IDS).

Therefore, at this point in time when the market for IDS products is growing, there is the need to develop new concepts and formulas for the risk assessments of IDS products.

As a result, in this study, I reviewed risk analysis concepts and formulas, analyzed the difficulties associated with using existing concepts and formulas for the assessment of IDS products. By examining the complexities of the networked environments in which the IDS operates, I illustrated (Figure 11) how the new concept - *Cascading Threat Multiplier (CTM)* can be used to calculate the SLE.

The CTM factors in the importance of other critical assets tied (re: networked) to the specific asset being analyzed in the Single Loss Expectancy (SLE) calculation and provides the analytical framework to closely scrutinize the assets under an organization's control, assign more comprehensive valuations to those assets, and to more accurately measure the impact that compromising of these assets could have on the organization.

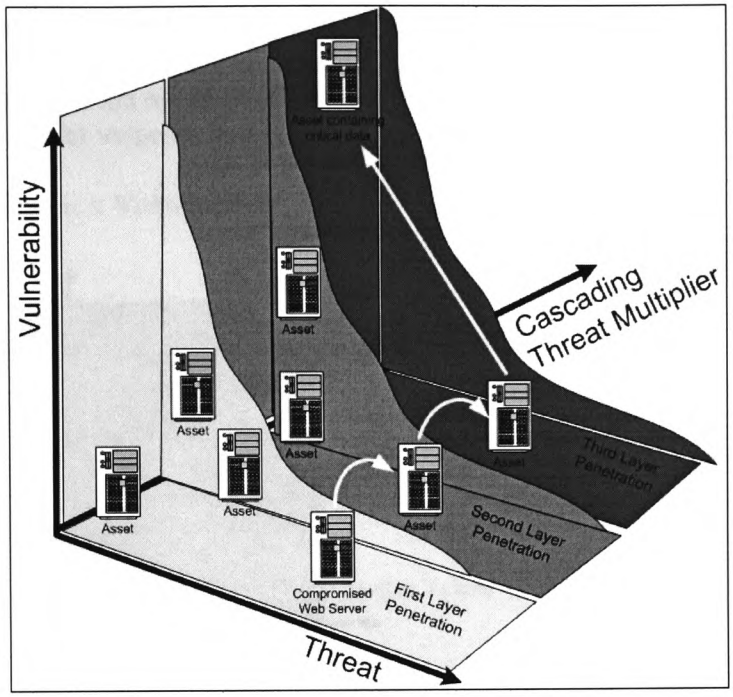


Fig. 11: Cascading threat multiplier Concept.

Therefore, this study contributes to the body of knowledge by illustrating new concepts and formulas and providing the analytical framework that can be used to conduct an accurate risk analysis of networked security (IDS) products.

4.2 The Impact of IDS Deployment Technique on Threat Mitigation

Brewer [47] states that the measure of risk can be determined as a product of threat, vulnerability and asset values:

Risk = Asset x Threat x Vulnerability.

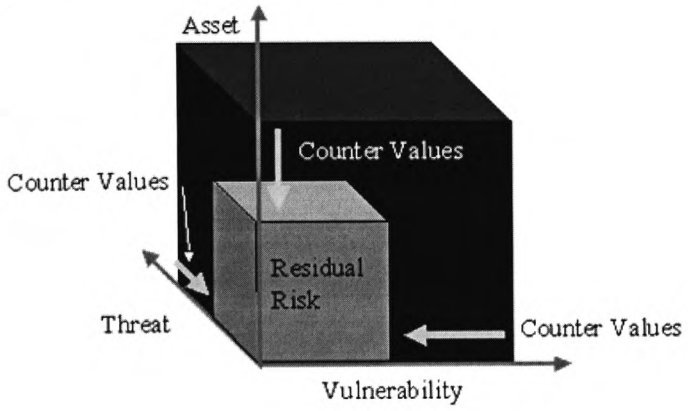


Fig: 12.Risk as a function of asset value, threat and vulnerability [47].

Further, he contends that the risk elements and their corresponding countermeasures for a specified system can best be visualized with a cuboid (Figure 12). In this case, the system has an initial level of risk before any countermeasures are applied. And countermeasures, assuming that their values are assigned by the same parameters that are used for threat, vulnerability and asset valuation, can reduce risk, i.e. by reducing threat (e.g. with locked doors, IDSes), reducing vulnerability (e.g. with awareness, patches, hot fixes) or reducing asset value (e.g. with encryption). After calculating the results from each combination of threat, vulnerability, asset and countermeasure the residual risk is determined [45]. Here the impact element is covered in asset value, the likelihood in threat and vulnerability values.

From the above, we conclude that threat mitigation depends on how the effectiveness of the applied countermeasures.

From a risk analysis perspective, in this study, we investigated the impact of deployment techniques on the IDS effectiveness in mitigating threat. This was accomplished by applying financial risk calculations to demonstrate the value of deploying IDS with different supporting procedures.

Two deployment techniques were considered: proactive vs. reactive. The merits of each were considered. By examining the Annual Loss Expectancy (ALE) in the case study scenario, we analyzed how the ALE variables are affected by each technique. Ultimately, we established that the proactive technique results into a higher ROI.

It needs to be noted that this study differs from a previous study [22] because the techniques considered deals with how the IDS is configured. Details of the proactive and reactive techniques are discussed in debt in the publication.

The results of this study demonstrate that in a reactive deployment, where personnel must be engaged to respond to each event, the risk exposure factors decreases. Equally, there will be similar benefit in a proactive deployment and in addition, the Annual Rate of Occurrence (ARO) will be reduced.

The contribution of this study to the existing body of knowledge is that we opened a new way of using the IDS configurable parameter to improve its performance and the ROI.

Subhypothesis Supported

The results of this study validate subhypothesis (3) i.e. "The manner in which the IDS is managed affects its performance effectiveness and ROI."

4.3 The Effect Of Intrusion Detection Management Methods On The Return On Investment

Prior to the procurement and deployment of new technologies, most organizations engage in a cost-benefit analysis to evaluate the economic or financial benefits of the new technologies or products they are implementing.

Thus, many an organizations investment decision will hinge on the ability to demonstrate a positive return on investment (ROI). For network security devices, the ROI has traditionally been difficult to quantify, in part because it is difficult to calculate risk accurately due to the complex issues involved in the analysis of networked environments. And also, business-relevant statistics regarding security incidents are not always available for consideration in analyzing risk.

For IDS implementations, there are no clearly established guides on effective management methods from the ROI perspective. Therefore, the intention behind this study is practical and grounded in real world challenges, which include:

1. Developing a risk analysis methodology that can be used as the framework to determine cost-effective management decisions of IDS implementations,
2. Establishing the technique that can be used to determine the viability of different management approaches and how they affect the IDS ROI.

In this study, I examined how management methods affect the IDS ROI. To do this, I used the risk equations introduced in Section 4.1 to performed ROI calculations for IDS implementations under different management schemes. For this, three ROI scopes were created under two management schemes (Table 1). Three possible scenarios were used to develop a possible method of reasoning about IDS ROI.

The result of the study (Table 1) demonstrates the benefits (as reflected in the reductions in the values for the ARO) of a better IDS management. The overall effect is visible in the increase in the ROI values for the IDS deployment for both the single in-house support and MSSP support schemes. Also, the study provided concrete proof that selection of any management method affects the implementation costs.

Scenario	ROI Scope	AV	EF	UEA	EFS	CTM	SLE	ARO	ALE1 (no IDS)	ALE2 (w/ AR)	ALE3 (w/ AR & IR)	Single Support ROI $	Single Support ROI %	MSSP Support ROI $	MSSP Support ROI %
One	no IDS	$2,000	75%	$20,000	75%	8.5	$12,750	3	$38,250	N/A	N/A	N/A	N/A	N/A	N/A
	IDS w/ Auto-Response	$2,000	75%	$20,000	75%	8.5	$12,750	1.5	$38,250	$19,125	N/A	-$64,092	-77%	-$25,092	-57%
	IDS w/ Auto-Response & Incident Response	$2,000	56%	$20,000	56%	6.6	$7,453	1.5	$38,250	N/A	$11,180	-$56,147	-67%	-$17,147	-39%
Two	no IDS	$20,000	50%	$50,000	50%	2.3	$22,500	2	$45,000	N/A	N/A	N/A	N/A	N/A	N/A
	IDS w/ Auto-Response	$20,000	50%	$50,000	50%	2.3	$22,500	1	$45,000	$22,500	N/A	-$60,717	-73%	-$21,717	-49%
	IDS w/ Auto-Response & Incident Response	$20,000	38%	$50,000	38%	1.9	$14,531	1	$45,000	N/A	$14,531	-$52,748	-63%	-$13,748	-31%
Three	no IDS	$3,000	75%	$200,000	50%	34.3	$77,250	1	$77,250	N/A	N/A	N/A	N/A	N/A	N/A
	IDS w/ Auto-Response	$3,000	75%	$200,000	50%	34.3	$77,250	0.5	$77,250	$38,625	N/A	-$44,592	-54%	-$5,592	-13%
	IDS w/ Auto-Response & Incident Response	$3,000	56%	$200,000	38%	26.0	$43,875	0.5	$77,250	N/A	$21,938	-$27,905	-34%	$11,096	25%
WBS ROI with IDS Auto-Response (ROIt)									$160,500	$80,250	N/A	-$2,967	-4%	$36,033	81%
WBS ROI with IDS Auto-Response & Realtime Incident Response (ROIr)									$160,500	N/A	$47,648	$29,635	36%	$68,635	155%

Table 1: IDS ROI for different management schemes

The contribution to the body of knowledge on IDS management is the development of a model that can be used to determine the viability of different management approaches for IDS implementation.

Subhypothesis Supported

This result also supports the subhypothesis (3) i.e. "The manner in which the IDS is managed affects its performance effectiveness and ROI" and subhypothesis (4) i.e. "Cost-effective implementation approaches will lead to a positive ROI."

4.4 Cost Effective Management Frameworks for Intrusion Detection Systems

The decision to deploy a security mechanism such as IDS is often motivated by the needs of security risk management. For some organizations, prior to implementation of an IDS product, the *cost-effectiveness* or *cost-benefit* trade-off both for the procurement and management is always a mission critical task. For the IDS to be cost-effective, it should cost no more than the expected level of loss from intrusions.

Generally speaking, cost-benefit analysis is conducted with cost models. In the business arena, cost benefit analysis incorporate the use of risk-adjusted cash flows in order to examine internal rate of return and maximum net present value figured as a percentage of information security expenditures. For the IDS, This entails conducting a cost-benefit analysis or the trade-offs of the basic cost components, which at the minimum include development cost, the cost of damage caused by an intrusion, the cost of manual or automatic response to an intrusion, and the operational cost, which measures constraints on time and computing resources.

A few theoretical cost models have been developed [48-53] for network intrusion detection systems. But, none seem to have been translated into practical usage for various reasons including the fact that in the current implementation of intrusion detection systems, cost value propositions are rare and the fact that many organizations are not educated about the cost-benefits of security systems and for some, analyzing site-specific cost factors could be very challenging as a result of the complexities of the networked environment in which they are deployed.

In essence, there is the need to move away from theoretical models into practicable/implementable models. To do this, we must first formulate the frameworks by applying a risk analysis procedure to select sensitive data/assets and create a cost matrix for each intrusion. This will then be used to develop implementable models.

Therefore, in this case study, we analyzed the factors that impact IDS implementation costs. We then discuss the different cost components including network and infrastructure-based costs. Using the two management methods: proactive and reactive (Table 2), we proposed the management frameworks that can be used to develop practicable models that can be used to calculate the cost-effectiveness of each IDS implementation.

Method	System actions	Personnel actions	Follow up information
Reactive	Log -> Alert ->	Respond -> Analyze ->Eradicate	Forensics and Evidence
Proactive	Respond - > Log -> Alert	Analyze -> Eradicate if necessary	Forensics and Evidence

Table 2: Proactive and Reactive Management Methods.

Our contributions to the body of knowledge is that we developed the frameworks that can be used develop models to support effective decision-making about which techniques are appropriate for the cost effective management of the IDS in a given environment.

In the big picture, selections of IDS implementation based on how well a strategy helps a company to perform in cost terms are preferable and will assist an IT officer to explain security mechanism selections more effectively to CEOs. In this case, the frameworks for the cost-benefit model (analysis) are effective in assessing network intrusion detection systems and can be used to periodically review the effectiveness of planned and implemented security controls to determine if they are doing what they are supposed to do, rather than creating additional problems.

Chapter 5: Analytical Studies

The studies reviewed in this Chapter are published under the following title:

"Intrusion Detection Systems in Large Organizations: Strategies for Effective Deployment and Sustenance." [Appendix 10]

5.1 Intrusion Detection Systems in Large Organizations: Strategies for Effective Deployment and Sustenance.

Acknowledging the need for IDS protection, and subsequently choosing the IDS that best fits the company's needs are important steps in the quest for overall information security. However, these steps only complete the initial stages of a thorough IDS implementation process. After selecting and purchasing the optimal IDS, a company must properly and efficiently deploy it throughout the organization.

The first step in a well-planned and thorough deployment should be to design an IDS strategy and then express it in the context of an IDS policy. This policy document serves as a guide for the implementation process, answering questions such as:

1. Will network traffic restrictions be tight or loose?
2. Who will be authorized to make changes to the IDS policy or configurations?
3. On which machines will an IDS installation be required?
4. How frequently will IDS logs undergo analysis?

The planning and coordination required in creating this policy will reinforce the communication between company management and security officials. At the same time, this will allow both organizational units to identify and resolve conflicts before they become obstacles to successful IDS deployment.

In the study, we performed a complex analysis of the IDS implementation in large setups (as reported in the literature and from the authors field experiences) to derive empirical arguments and fact that we used to formulate strategies for IDS implementation and performance enhancement. The argument is to use IT security policy and systems configurations to make the IDS more effective.

Thus, the strategies address the challenge before an organization about how to deal with the issue of setting the IDS to capture relevant data only. And for every organization, there are different expectations and, such that the default IDS settings usually need to be altered. Finding the perfect balance between a massive amount of data generation, which leads to an over-saturation of information, and a small amount of data generation, which may cause ineffective monitoring, can complicate a deployment. In general, a sophisticated IDS solution will require a sophisticated IDS configuration, so organizations must seek optimal strategies for thorough configuration development, tuning, and testing.

The propositions we have put forth here are effective lifecycle performance enhancement strategies for IDS procurement, implementation, management and maintenance. This entails:

(1) Detailing an organization's approach to intrusion detection in general and the

implementation policy that determines which strategy to employ, hence determine what can be done to help improve IDS performance. This could for instance stipulate the methods to monitor attacks. Possible options include:

1. to monitor for all attacks, regardless of what systems are prevalent in an organization; for example, looking for RPC exploits in a Microsoft environment. *This option would be more expensive since the volume of work would be large and the IDS effectiveness would be reduced.*

2. to monitor only for attacks that would be relevant to the network environment, such as configuring the NIDS to detect all Microsoft exploits in an all Microsoft environment. *This will reduce costs on network resources usage, personnel, etc.;*

3. to monitor all vulnerabilities for a particular service regardless of the environment, such as detecting all HTTP exploits in an IIS-only environment. *This would lead to a reduction in the IDS workload, the volume of data to be analyzed resulting to cost savings on personnel and network resources usage.*

(2) By optimizing systems configuration settings in affected operational areas. This requires taking a security policy as the basis for the configuration of the IDS components; and as the basis to optimize system configurations in order to make the IDS more responsive to these settings.

Subhypothesis supported.

This analytical work has provided the reasonable arguments, facts and propositions in support of the assertions of subhypothesis (5) that "System configurations settings play a role in IDS effectiveness" and subhypothesis (6) that "Tailoring the IDS function to be more consistent with local security policy improves the IDS effectiveness."

Chapter 6. Research and Development Studies

The studies reviewed in this Chapter are:

1. Intrusion Detection Challenges: charting the course for research and development. [Appendix 11]

2. Future Directions in the Development of Intrusion Detection Systems.[Appendix 12]

6.1 Intrusion Detection Challenges: charting the course for research and development

As with other security and monitoring products, intrusion detection systems functions as one element of a corporate security policy. Successful intrusion detection requires that a well-defined policy on IDS development be formulated to ensure that intrusions are handled according to corporate security policy guidelines.

Currently available IDS technologies face several technical and implementation challenges that threaten the IDS market share. Hence, the intrusion detection system technology requires considerable refinements to eliminate the weaknesses in currently available products. Some of the weaknesses that are considered short-term i.e. scalability, hierarchical reporting, and dynamic remote updates are already being addressed by vendors while the long-term weaknesses are being addressed through several ongoing research and development efforts worldwide.

In this study, we reviewed the different issues and problems associated with the IDS technology; and putting the issues in a research and development context, we proposed a roadmap of potential research topics and articulated the issues important to explore within each research topic.

The hope is that researchers can use this to navigate their research interests as they work to develop the most appropriate remedies to current design problems.

6.2 Future Directions in the Development of Intrusion Detection Systems

Current IDS products bring the ability to view network and system activity in real-time, identify unauthorized activity and provide a near-real-time automated response. IDS products also provide the ability to analyze today's activity in view of yesterday's activity to identify larger trends and problems. It is reasonable to expect IDS technology to revolutionize computer security efforts, by allowing real-time operational capability in controlling unauthorized activity in corporate cyberspace. IDS technology does not directly address other security issues such as identification/authentication, confidentiality, etc., though some of these technologies will be integrated with IDS in the near future.

Anticipating the effects of emerging IDS technologies, this research reviewed the pitfalls of commercially implemented IDS products and provided detailed technical discussions on several aspects embodying several research choices likely to facilitate high-quality product design.

We also reviewed the expectations revolving around what future IDS should look like and what it should accomplish to remain viable as an IT security technology.

Chapter 7. Conclusion

7.1 Conclusion

In this thesis, I hypothesized that the effectiveness of intrusion detection systems (IDS) can be improved by rethinking the way the IDS is managed and by adopting effective and systematic implementation approaches. To show the validity of this hypothesis, I have (in collaboration with Dr. Andrew Blyth and a few others) conducted research studies using different approaches: experimental studies, case studies, and analytical studies.

Results of these studies support the hypothesis.

I will now summarize the work related to each approach, discuss specific contributions and consider future work.

7.2 Summary

In the foundation studies on IDS concepts and approaches, we explored different IDS designs and implementation techniques, summarized the advantages and disadvantages of each intrusion detection approach and then analyzed the suitability of use of each in different environments. From empirically derived arguments we proposed effective methods that can be used to incorporate established engineering principles and standards into IDS design and suggested strategies to make IDS implementation seamlessly integral with enterprise security policies and standards.

Regarding the experimental studies, we developed methodologies for testing IDSes in switched and gigabit environments. The methodologies consists of general software-testing techniques, which we have adapted for the specific purpose of testing the IDS; and the misuse detection approach to evaluate the performance of the IDS against selected tunable parametric specifications under varying test conditions. With these methodologies we preformed tests to measure the IDS effectiveness against a wide range of environmentally desirable characteristics for a broad range of known intrusions.

Concerning the case studies, we reviewed current risk assessment concepts, techniques, and formulas, proposed new concepts for the risk assessment of IDS products, investigated the relationship between implementation techniques and threat mitigation, examined how management methods affect the IDS ROI, formulated the frameworks that can be used to determine the cost-effectiveness of IDS, and addressed the problems of bridging the gap between technical security solutions and the business need for the IDS.

In the analytical study, we analyzed and suggested how to optimize the settings of

systems configurations to enhance the IDS effectiveness; and proposed strategies that make the IDS function in accordance with local security policy in large-scale organizations.

As for future work, we culled from the various parts of our research studies a list of some of the most pressing issues and subjects matters pertinent to the IDS effectiveness as potential research topics.

7.3 Research Contributions

We made the following specific contributions to the security research community.

IDS Concepts and Approaches.

- We provided a systems-based description of intrusion detection technologies, analyzed the suitability of use of each approach for different environments and proposed a conceptual design approach and a technique for designing secure IDSes, which are guaranteed to be correct in the sense that a specified security criterion will not be violated if proper validation principles act correctly.

IDS Deployment Techniques.

- *We developed methodologies for testing IDSes in switched environments and; to evaluate the performance of different IDS products with different design architectures in different environments (Mbps vs. Gbps).*

- *We established the best techniques to deploy the IDS within a switched network environment. This could be used for organizations trying to justify and/or optimize their IDS deployment techniques in order to maximize the IDS effectiveness.*

- *Finally, we established empirical bandwidth limits for the selection of appropriate IDS technology/product in highly scalable environments where Gigabit architectures are used. This could serve as benchmarks to determine when multiple 100Mbps IDS sensors can be more effective and thus preferentially used instead of a single Gigabit IDS sensor in environments with bandwidths exceeding 100Mbps or Gigabit environments.*

Implementation/Management Methods.

- *We demonstrated the correctness and the application of a new concept in asset valuation and risk analysis of IDSes – "the Critical Threat Multiplier (CTM)."*

- *We developed a model that can be used to determine the viability of different management approaches and how they affect the IDS ROI; and how to accurately calculate the ROI for IDS implementations.*

- *We developed a model that can be used to improve the accuracy value when calculating the ROI for IDS implementations.*

- *We conceived of strategies and approaches to support effective decision-making about which techniques are appropriate for the cost effective management of the IDS in a given environment.*

- *We opened a new way of estimating IDS effectiveness by applying financial risk calculations to demonstrate the value of deploying IDS with different supporting procedures. In this case, we demonstrated the real potential impacts of deploying IDS technologies from a business setting, which presents a good balance to the IT security community.*

- *We developed the frameworks that can be used to analyze site-specific cost factors for IDS implementation.*

Systems Configuration and Security Policy.

- *We proposed methods that take a security policy as the basis for the configuration of the IDS components; and effectiveness enhancement strategies.*

- *We conceived of and proposed several techniques for optimizing system configuration settings to make the IDS more responsive and effective to these settings.*

Research and Development Studies.

- *We presented detailed technical discussions on several aspects embodying several research choices likely to facilitate high-quality product design and provided a tangible reflection to some of the needs arising from the pitfalls of the current designs, and suggested the trends likely to bring radical changes in the meaning and modes of IDS implementation in the years ahead.*

7.4 Future Work

We have culled from the likely scenarios sketched in various parts of our research studies the following list of some of the most pressing issues and subject matters to be pursued in future research:

1. ***Excessive alerts:*** The sheer volume of INFOSEC device alerts makes security management a time-consuming and therefore expensive effort.

2. ***Alert Management tool:*** Among the most pressing problem for active research is the development of technologies to manage and interpret security relevant alert streams produced from an ever-increasing number of INFOSEC devices.

3. ***Algorithms***: Domain expertise is not widely available that can interpret and isolate high threat operations within active and visible Internet-connected networks. In an environment where thousands (or tens of thousands) of INFOSEC alarms may be produced daily, it is important to understand redundancies in alert production that can simplify alert interpretation. Equally important are algorithms for prioritizing which security incidents pose the greatest administrative threats.

4. ***Information management method:*** In managing INFOSEC devices, it is difficult to leverage potentially complementary information produce from heterogeneous INFOSEC devices. As a result, security relevant information that, for example, is captured in a firewall log, is typically manually analyzed in isolation from potentially relevant alert information captured by IDS, Syslog, or other INFOSEC alert source.

5. ***Data sets***: Better data sets are necessary for better calculation of metrics in future evaluations and to further research. Datasets will need to take on new forms such as specifications and tools for created attack and background traffic in ones own environment so that IDS developers can explore use of new and different inputs for their systems.

6. ***Anomaly-based detection approach:*** Generally speaking, there seems to be much interest in going back to the anomaly-based approach of years ago without really understanding the value of what has been accomplished with the misuse detection approach. Thus, the industry is likely to move much faster to address the anomaly-based approach because of the successes and lessons learned from the misuse approach.

7. ***Expert-based approaches:*** A large number of IDS researchers are working on expert-based approaches because those are technically more interesting and are more likely to really evolve into something useful in the long run. The big gap is that the research tends to also ignore the "real security equals network management" problem and builds systems that are hard to manage, don't have intuitive user interfaces (or documentation) or that are cumbersome to use. It is likely that the good ideas from the R&D systems will wind up in commercial products. This will be the right research direction since good ideas, not products,

come from research.

8. ***Data correlation***: It could be successfully argued that the future of IDS lies in data correlation research. The IDS of tomorrow will produce results by examining input from several different sources.

9. ***Audit trails***: Research to determine what kinds of information should be in audit trails, and when such data needs to be collected to optimally drive any intrusion detection system will be critical in defining the architecture of data mining technologies.

10. ***Storage format:*** Research to determine the best structure/storage formats for audit data so that it can be quickly processed without taking up huge amounts of storage will aid data mining architectural designs.

11. ***Software automation:*** Exploring how to define policy in a consistent and meaningful way such that it can be expressed in software for automated comparison and detection of intrusions and internal misuse is a viable research field.

12. ***Reference model:*** There is a need to develop a reference model for IDS design as any meaningful design should take a queue from a standard reference model just as the one done by Christopher Schuba on a formal reference model for firewalls.

Research on these issues would enlighten the future development of IDSes, and their role in devising improved public policy and planning based on the best available information.

References

[1] P. Dorosz, P. Kazienko, Systemy wykrywania intruzów, VI Krajowa Konferencja Zastosowan Kryptografii ENIGMA 2002, Warsaw 14-17 05.2002. , p. TIV 47-78, http://www.enigma.com.pl/konferencje/vi_kkzk/index.htm.
[2] James P. Anderson, "Computer Security Threat Monitoring and Surveillance", *Technical report, James P. Anderson Co., Fort Washington, PA., April 1980.*
[3] Dorothy E. Denning, "An intrusion-detection model", *IEEE Transactions on Software Engineering, vol. SE-13, pp. 222-232, February 1987.*
[4] Teresa Lunt et al., "IDES: The enhanced prototype", *Technical report, SRI International, Computer Science Lab, October 1988.*
[5] D. Anderson, T. Frivold, A. Valdes, "Next-generation intrusion detection expert system (NIDES)", *Technical report, SRI-CSL-95-07, SRI International, Computer Science Lab, May 1995.*
[6] S. E. Smaha, "Haystack: An Intrusion Detection System", *Proceedings of the IEEE Fourth Aerospace Computer Security Applications Conference, Orlando, FL., December 1988.*
7] M Sebring et al., "Expert systems in intrusion detection: A case study", *Proceedings of the 11th National Computer Security Conference, Baltimore, MD., October 1988.*
[8] H. S. Vaccaro, G. E. Liepins, "Detection of anomalous computer session activity", *Proceedings of the 1989 Symposium on Research in Security and Privacy, Oakland, CA., May 1989.*
[9] J. R. Winkler, W. J. Page, "Intrusion and Anomaly Detection in Trusted Systems", *Proceedings of the Fifth Annual Computer Security Applications Conference, Tucson, AZ., December 1989.*
[10] L. T. Heberlein et al., "A network security monitor", *Proceedings of the IEEE Symposium on Research in Security and Privacy, Oakland, CA., May 1990.*
[11] K. Jackson, D. DuBois, C. Stallings, "An expert system application for network intrusion detection", *Proceedings of the 14th Department of Energy Computer Security Group Conference, 1991.*
[12] S. R. Snapp et al., "A system for distributed intrusion detection", *Proceedings of the IEEE COMPCON 91, San Francisco, CA., February 1991.*
[13] Mark Crosbie, Gene Spafford, "Defending a Computer System using Autonomous Agents", *Technical report No. 95-022, COAST Laboratory, Department of Computer Sciences, Purdue University, March 1994.*
[14] S. Staniford-Chen, S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. Levitt, C. Wee, R. Yip, D. Zerkle, "GrIDS -- A Graph-Based Intrusion Detection System for Large Networks", *The 19th National Information Systems Security Conference, Baltimore, MD., October 1996.*
[15] Ross Anderson, Abida Khattak, "The Use of Information Retrieval Techniques for Intrusion Detection", *Proceedings of RAID '98, Louvain-la-Neuve, Belgium, September 1998.*

[16] Richards K, "Network Based Intrusion Detection: a review of technologies", *Computers & Security*, 18 (1999) 671-682.
[17] H. Debar, M. Dacier, A. Wespi, "Towards a taxonomy of intrusion-detection systems", *Computer Networks* 31, 1999, pages 805-822.
[18] E. Lundin, E. Jonsson, Survey of research in the intrusion detection area, Technical report 02-04, Department of Computer Engineering, Chalmers University of Technology, Göteborg January 2002,
[19] C. Krügel, T. Toth, Applying Mobile Agent Technology to Intrusion Detection, ICSE Workshop on Software Engineering and Mobility, Toronto May 2001.
[20] C. Krügel, T. Toth, Distributed Pattern Detection for Intrusion Detection, Conference Proceedings of the Network and Distributed System Security Symposium NDSS '02, 2002,
[21] J.S. Balasubramaniyan, J.O. Garcia-Fernandez, D. Isaco, E. Spafford, D. Zamboni, An Architecture for Intrusion Detection using Autonomous Agents, 14th IEEE Computer Security Applications Conference ACSAC '98, December 1998, pages 13-24.
[22] D.J. Ragsdale, C.A. Carver, J.W. Humphries, U.W. Pooh, Adaptation techniques for intrusion detection and intrusion response systems, Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 2000, pages 2344-2349.
[23] Iheagwara C. and Blyth A, "Evaluation of the performance of IDS systems in a switched and distributed environment," *Computer Networks*, 39 (2002) 93-112.
[24] M. Esmaili, R. Safavi-Naini, and J. Pieprzyk, "Computer Intrusion Detection: A Comparative Survey," Tech. *Rep. 95-07*, Center for Computer Security Research, University of Wollongong, May 1995.
[25] T. F. Lunt, "A Survey of Intrusion Detection Techniques," *Computers & Security*, vol. 12 (4), 1993, pp. 405-418.
[26] T. Lane, "An Application of Machine Learning to Anomaly Detection," Tech. *Rep. 97-03*, COAST Laboratory, Department of Computer Science, Purdue University, February 1997.
[27] K. Ilgun, "USTAT: A real-time intrusion detection system for UNIX," *Proc. IEEE Symp. on Research in Security and Privacy*, Oakland, CA, May 24-26, 1990, pp. 16-28.
[28] S. Kumar and E. H. Spafford, "A Pattern Matching Model for Misuse Intrusion Detection," *Proc. 17th National Computer Security Conf.*, Baltimore, MD, October 11-14, 1994, pp. 11-21.
[29] T. D. Garvey and T. F. Lunt, "Model based Intrusion Detection," *Proc. 14th National Computer Security Conf.*, Washington, DC, October 1-4, 1991, pp. 372-385.
[30] C. Ko, M. Ruschitzka, and K. N. Levitt, "Execution Monitoring of Security-Critical Programs in Distributed Systems: A Specification-based Approach," *Proc. of IEEE Symposium on Security and Privacy Conf.*, May 4-7, 1997, pp. 175 - 187.
[31] R. Buschkes, M. Borning, and D. Kesdogan, "Transaction-based Anomaly Detection," *Proc of Workshop on Intrusion Detection and Network Monitoring*, Santa Clara, CA, April 9-12, 1999.
[32] M. Crosbie and E. H. Spafford, "Active Defense of a Computer System using

Autonomous Agents," *Tech. Rep. 95-008*, COAST Laboratory, Computer Science Dept., Purdue University, February 1995.
[33] J. Balasubramaniyan, J. O. Garcia-Fernandez, D. Isacoff, E. H. Spafford, and D. M. Zamboni, "An Architecture for Intrusion Detection using Autonomous Agents," Tech. *Rep. 98-05*, COAST Laboratory, Department of Computer Science, Purdue University, May 1998.
[34] G. B. White, E. A. Fisch, and U. W. Pooch, "Cooperating Security Managers: A Peer-based Intrusion Detection System," *IEEE Network*, vol. 10 (1), 1996, pp. 20-23.
[35] G. B. White and U. W. Pooch, "Cooperating Security Monitors: Distributed Intrusion Detection Systems," *Computers and Security*, vol. 15 (5), 1996, pp. 441-450.
[36] Meadows C.1995. Formal Verification of Cryptographic Protocols: A Survey, *Advances in Cryptology, Proceedings of ASIACRYPT '94,* 133-150, Springer Verlag.
[37] Heintze N., Tygar J. 1995. A Model for Secure Protocols and their Compositions In the *Proceedings of the 1994 IEEE Symposium on Security and Privacy*, IEEE Computer Society Press.
[38] Rudolph C. 1998. A Formal Model for Systematic Design of Key Establishment Protocols, *ACISP'98 Proceedings of the Third Australasian Conference on Information Security and Privacy*, 332-343, Springer Verlag.
[39] Buttyan L., Staamann S., Wilhelm U. 1998. A Simple Logic for Authentication Protocol Design. In *Proceedings of the IEEE Computer Security Foundations Workshop XI*, 153-162, IEEE Computer Society Press
[40] Gollmann D. 1996. What do we mean by Entity Authentication, In the *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, 46- 54, IEEE Computer Society Press.
[41] P. A. Porras and A. Valdes, Live Traffic Analysis of TCP/IP Gateways, In: *Internet Society's Networks and Distributed systems Security Symposium*, March 1998.
[42] Kevin Richards, Network Based Intrusion Detection: a review of technologies, Computers & Security, 18 (1999) 671-682.
[43] Banga, G., and Druschel, P., Measuring the capacity of a Web server, Proceedings of USENIX Symposium on Internet Technology and Systems, California, USA, December 1997.
[44] Almeida, J., and Cao, P., Measuring Proxy performance with the Wisconsin Proxy Benchmark, Technical document, Department of Computer Science, University of Wisconsin-Madison, USA, July 1997
[45] Brewer, D. "Risk Assessment Models and Evolving Approaches." Gamma Secure Systems, Diamond House, Frimley Road, Camberley, Surrey, March 22, 2002.
[46] Harris, S. "CISSIP Certification Guide.", McGraw-Hill/Osborne, 2001, pp72-91.
[47] Brewer, Dr. David. "Risk, Security and Trust in the Open World of E-Commerce." May 1999. URL: http://www.itsecurity.com/papers/p35.htm (22 March 2002).
[48] Lee W. et al, "Toward Cost-Sensitive Modeling for Intrusion Detection and Response", North Carolina State University, 1999.
[49] Stolfo S. et al, "Cost-Based Modeling for Fraud and Intrusion Detection Results from the JAM Project", Technical Report, Columbia University.
[50]. Butler, S. A. Security Attribute Evaluation Method: "A Cost-Benefit Approach". *In*

Proceedings of the International Conference on Software Engineering, Orlando, Florida, 2002.
[51] Wei H. et al, "Cost Benefit Analysis for Network Intrusion Detection Systems", In: *Proceedings of the CSI 28th Annual Computer Security Conference,* Washington, D.C., October 2001.
[52] Irvine C. et al, "Toward a Taxonomy and Costing Method for Security Metrics", In: *Proceedings of the Annual Computer Security Applications conference*, Phoenix, AZ, Dec. 1999
[53] Cohen et al, "A Preliminary Classification Scheme for Information System Threats, Attacks, and Defenses; A Cause and Effect Model; and Some Analysis Based on That Model", *Sandia National Laboratories*, September, 1998.

Appendices of Publications

Appendix 1: "A Comparative Analysis of Intrusion Detection Systems." In: Proceedings of the FIRST 15th Annual Computer Security Incident Handling Conference The Westin, Ottawa Ontario, Canada, June 22-27, 2003.

Appendix 2: "Security problems and the interaction of security policies in the design and implementation of IDS in enterprise networks." In: Proceedings of the FIRST 15th Annual Computer Security Incident Handling Conference. The Westin, Ottawa Ontario, Canada, June 22-27, 2003.

Appendix 3: "The impact of security layering on end-to-end latency and system performance in switched and distributed e-business environments." Computer Networks Journal, Volume 39-5.

Appendix 4: "Evaluation of the Performance of ID Systems in a Switched and Distributed Environment: The RealSecure Case Study." Computer Networks Journal, Volume 39 –2.

Appendix 5: "A Comparative Experimental Evaluation Study of Intrusion Detection System Performance in a Gigabit Environment." Journal of Computer Security, Volume 11 (2003) 1-33.

Appendix 6: "Towards an Effective Risk Assessment Methodology: Factoring in Novel Concepts For Assessing Intrusion Detection Systems in Complex Infrastructures" Computer Security Journal, Volume XIV, Number 2, 2003.

Appendix 7: "The Impact of IDS Deployment Technique on Threat Mitigation." In: Proceeding of the International Conference on Industrial Engineering and Engineering Management (IE&EM'2003), Shanghai, China, on December 6-8 2003.

Appendix 8: "The Effect of Intrusion Detection Management Methods on the Return on Investment." Computers & Security Journal, Volume 23, 213-228.

Appendix 9: "Cost Effective Management Frameworks for Intrusion

Detection Systems." Journal of Computer Security, Volume 12, Number 5, 2004, pp. 777-798.

Appendix 10: "Intrusion Detection Systems in Large Organizations: Strategies for Effective Deployment and Sustenance." In: Proceeding of the International Conference on Industrial Engineering and Engineering Management (IE&EM'2003), Shanghai, China, on December 6-8 2003.

Appendix 11: "Intrusion Detection Challenges: charting the course for research and development." In: Proceeding of the International Conference on Industrial Engineering and Engineering Management (IE&EM'2003), Shanghai, China, on December 6-8 2003.

Appendix 12: "Future Directions in the Development of Intrusion Detection Systems." The Information Systems Control (ISACA) Newsletter, May 2003.

Appendices (Publications)

Appendix 1

"A Comparative Analysis of Intrusion Detection Systems" In: Proceedings of the FIRST 15th Annual Computer Security Incident Handling Conference The Westin, Ottawa Ontario, Canada, June 22-27, 2003

www.first.org

num Sponsor:

Government of Canada
Office of Critical Infrastructure Protection and Emergency Preparedness

Gouvernement du Canada
Bureau de la protection des infrastructures essentielles et de la protection civile

Binder Sponsor:

BT

A Comparative Analysis of Intrusion Detection Systems

Charles Iheagwara

Abstract

This paper examines intrusion detection systems and provides a system-based analytical comparison of the leading implementation approaches, techniques and systems. Intrusion detection systems detect attacks that attempt to compromise the integrity, confidentiality, or availability of a resource. In particular, this paper provides a systems-based description of intrusion detection technologies.

Keywords: intrusion detection, computer security

1.0 INTRODUCTION

An Intrusion detection system (IDS) is a security system that monitors computer systems and network traffic and analyzes that traffic for possible hostile attacks originating from outside the organization and also for system misuse or attacks originating from inside the organization.

Intrusion detection systems evolved due to the lack of intrusion prevention systems and the need to address the following issues:

- It is impossible to build a completely secure system in today's software development environment because the programming languages and operating systems used for development and implementation introduce a number of security flaws. These security flaws are difficult to detect and intruders can use these flaws to bypass existing security mechanisms. Figure 1 provides a partial taxonomy of these security flaws [1].

- The enormous installed base of operating systems and applications ensure that the replacement of existing systems with a secure system will require a transition period measured in decades.

- Existing cryptographic systems are not completely secure and have exploitable weaknesses for a determined and resourceful intruder. The best cryptographic system offers no protection against lost or stolen keys or poorly chosen passwords.

- There is an inverse relationship between the level of system security and user efficiency. As system security increases, user efficiency decreases. A completely secure system, with existing security techniques, is practically unusable.

- Finally, a secure system may still be vulnerable to an insider misusing their privileges.

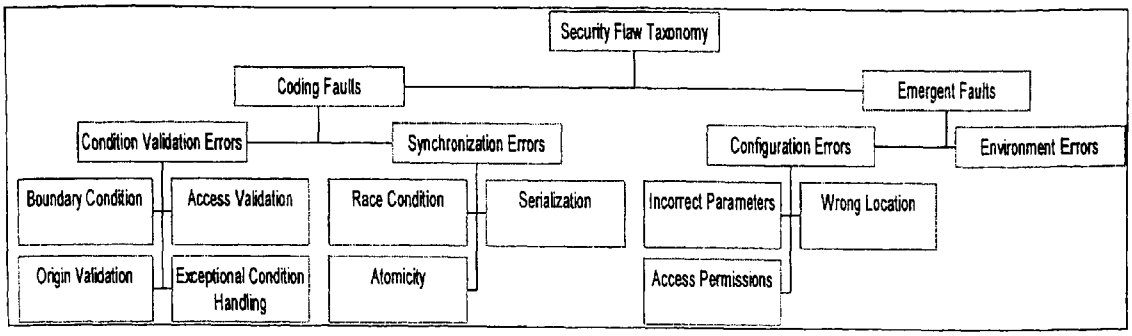


Figure 1: Partial Taxonomy of Security Flaws

IDS, much like the security industry itself, have grown rapidly over the past few years. These tools have become essential security components - as valuable to many organizations as a firewall. However, as in any environment, things change. As networks and crackers evolve and grow rapidly, demanding that security tools keep up, the IDS faces several daunting but exciting challenges in the future and are sure to remain one of the best weapons in the arena of network security.

One of the major reasons for the growth of IDS products and technologies is the advent of Internet connectivity, threats, and financial incentive for attackers. The advent of the World Wide Web has led to increased interconnectivity, increased demands for network services, and increased threats.

The recent CSI-FBI survey [2] of 503 American organizations validated the continued concerns of business leaders today with doing business in the electronic era. Of the 503 organizations surveyed, 90% detected a security breach of their information systems and 80% experienced financial losses as a result of breaches. While internal threats remain a top priority, 40% cited breaches from outside their organization. Additionally, 85% experienced viruses and 74% stated their Internet connection was most frequently targeted. The most signification piece of data from this survey indicates that 90% of these respondents have a Web site, 90% have firewalls and antivirus programs and 100% conduct business electronically in some fashion.

The statistics in the survey points to a notable trend, not necessarily the percentages, but simply that 100% of those surveyed are conducting business electronically and 90% of them have firewalls and antivirus, yet 90% reported system breaches. Protecting information systems today must be done in a layered process, which includes technology and human analysis. As the CSI-FBI survey revealed, most companies have already deployed firewalls and antivirus programs, and many are moving aggressively towards acquiring Intrusion Detection Systems (IDS), a security system that monitors computer systems and network traffic and analyzes that traffic for possible hostile attacks originating from outside the organization and also for system misuse or attacks originating from inside the organization.

The reminder of the paper is organized as follows. In Section 2 we discuss the various IDS implementation approaches and present a classification of the techniques in Section

3. A descriptive analysis of IDS systems is given in Section 4 and the conclusions are given Section 5.

2.0 IDS APPROACHES

There are three broad approaches for intrusion detection: anomaly detection, misuse detection, and specification-based detection. In practice, none of the three are sufficient for a robust intrusion detection system - a combination of two or all three approaches is necessary.

2.1 Anomaly Detection

In the anomaly detection model, this is realized by detecting changes in the patterns of utilization or behavior of the system performs detection. Building a model that contains metrics derived from normal system operation and flagging as intrusive any observed metrics that have a significant statistical deviation from the model perform it. In this case, the user profile is a collection of metrics such as average CPU load, number of processes, login time, or number of network connections that characterizes user activity. Threshold levels are set for these metrics, and activities above these thresholds are characterized as intrusions [3].

Because intrusions are a subset of anomalous activity, it is possible to flag anomalous activity as intrusive when it is not (*false positive*), or to ignore intrusive behavior because the anomaly detection system does not consider it abnormal (*false negative*).

Figure 2a provides a visual explanation of anomaly detection systems and the relationship between intrusions, false positives and false negatives in anomaly detection systems. Set a represents the event space that the anomaly detection system believes is so anomalous that it is intrusive. Set I represents actual intrusions. $A \cap I$ is the set of activities reported as intrusions. I - A is the set of false negative while A - I is the set of false positives. Typically, anomaly detection systems generate relatively few false negatives but have the potential for generating a large number of false positives (e.g. $|A-I| > |I-A|$).

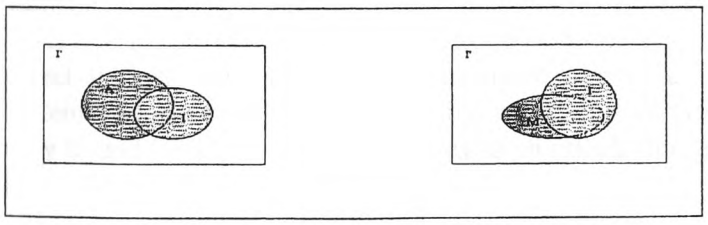


Figure 2: (a) Anomaly Detection and (b) Misuse Detection Event Space

There are a number of compromises involved in building anomaly detection systems. The effectiveness of the system is dependent on the number of metrics monitored and the frequency at which these metrics are monitored. The accuracy of the anomaly detection increases as the number of metrics and frequency of monitoring increases. The system requirements of the anomaly detection system likewise increase requiring a compromise between system performance and model accuracy.

Anomaly detection systems have a number of advantages and disadvantages. Because anomaly detection systems do not attempt to detect or classify specific attacks, new attacks can be detected without reprogramming. Furthermore, anomaly detection systems can be trained to accurately model users and can adapt to user changes in work practices over time. Unfortunately, anomaly detection systems can impose a high overhead on system performance. As the user model becomes more complex and hopefully accurate, the anomaly detection system must monitor and interrelate more metrics.

2.2 Misuse Detection

Detection is performed in the misuse detection model by looking for specific patterns or sequences of events representing previous intrusions (i.e. looking for the "signature" of the intrusion. It is a knowledge-based technique and only known intrusions can be detected. This is the more traditional ID technique, which is usually applied, in for instance the anti-virus tools.

Misuse detection systems can report false positives and negatives like anomaly-based systems. If a signature matches normal user activity as well as intrusive behavior, then a false positive is reported. If a new attack is developed for which an attack signature does not exist, then a false negative will occur.

Figure 2b provides a visual explanation of misuse detection systems and its relationship to intrusions, false positives, and false negatives. Misuse detection is based on the accuracy of its attack signatures, which must be very specific. If intruders use attacks unknown to the misuse detection system, a false positive is generated. A false positive occurs only when normal user activity matches an attack signature and is not an attack. Typically, misuse detection systems generate very few false positives but have the potential for generating a large number of false negatives (e.g. $|I-M| > |M-I|$).

Misuse detection systems have a number of advantages and disadvantages. Because attack signatures can be targeted to specific attacks, the number of false positives can be reduced significantly. This unfortunately leads to an increase in the number of false negatives, as intrusions must match the attack signature. This also introduces a period of vulnerability between when a new attack is developed and when an attack signature is generated for the attack. Anomaly detection systems do not have this vulnerability as they detect anomalous behavior and do not need a specific attack signature. Misuse detection systems also have difficulty handling significant variations of an attack. It is relatively easy to add commands or procedures to an attack that do nothing but obscure the actual attack [4].

2.3 Specification-based Detection

Specification-based detection focuses on expected system behavior instead of user activity. System behavior is formally specified for all circumstances and a profile is developed. The system is then monitored and all its actions are compared against the profile; system behavior that is not specified as correct is flagged as an intrusion [5].

A possible implementation of specification-based detection system is the use of a special policy specification language. This specification language would stipulate security policy by assigning access privileges to each file in the system.

Specification-based detection systems can have false negatives but if system behavior is specified accurately, there are no false positives. False negatives can occur when the system specification does not cover all possible system states. False positives can only occur if the system behavior is not specified accurately.

Specification-based detection systems have a number of advantages and disadvantages. One advantage of specification-based systems is that the number of false positive and negative reports can be minimized through accurate and complete specification of the system state. Additionally, like the anomaly-based approach, attacks can be detected even if they have not been previously encountered. The principal disadvantage is the fundamental requirement to specify explicitly security policy. A complete specification of a system would require a great deal of time and expertise. If the system was dynamic, maintaining an accurate specification could be very time-consuming.

2.4 Comparison of IDS Approaches

IDS approaches address different types of intruders. Anomaly systems detect marauders better than misuse systems under the assumption that the marauder's usage pattern is significantly different from the user. Misuse systems can detect misfeasors while anomaly systems are generally ineffective. Misfeasors can train the anomaly detection system to consider intrusive behavior as "normal" for the user over time. Both anomaly and misuse have limited utility against a clandestine attacker. Once an intruder has supervisory permission on a system, detection becomes very difficult as the skilled clandestine attacker can alter all logging and audit mechanisms to cover his intrusion. No single IDS approach is sufficient for detecting all intrusions. Instead, a combination of approaches is necessary to protect against different types of attacks.

Patterns of usage also influence the effectiveness of a particular IDS approach. If the users are in a production environment where they repeatedly use a limited subset of commands in a particular order, anomaly detections work extremely well. If the users use the system infrequently or have no set pattern of usage, then misuse detection systems tend to outperform anomaly detection systems.

Table 1 summarizes the advantages and disadvantages of each intrusion detection approach. Most IDS implement a combination of approaches to balance the advantages and disadvantages of each approach.

3.0 CLASSIFICATION OF TECHNIQUES

There are a number of classification techniques that can be used within intrusion detection approaches. These techniques classify events as either intrusive or normal and include statistical analysis, predictive patterns, state transition, expert systems, neural networks, machine learning, pattern matching, graph-based, and model-based approaches. This section will examine these techniques.

Approach	Advantages	Disadvantages
Anomaly	Can detect new attacks without reprogramming. Few false negatives.	Potential for many false positives. Insiders can train user model to classify intrusive behavior as normal.
Misuse	Few false positives	Potential for many false negatives due to vulnerabilities to unknown attacks. Easy to obscure attack
Specification	Potentially no false positives	Very difficult to specify all system states.

Table 1: **Comparison of IDS Approaches**

3.1 Statistical Analysis

Statistical analysis is an anomaly detection technique that uses differences in the volume and type of audit data to detect intrusions. This is one of the earliest forms of intrusion detection and has been used in a large number of IDS. There are two forms of statistical analysis used for intrusion detection: threshold detection and the profile-based approach [6].

3.1.1Threshold detection

Threshold detection uses summary statistics on system and user activities to detect intrusions. The parameters of a threshold detection system are: what activity should the IDS measure and monitor; how often should the IDS perform analysis on this measurement; and what level of activity is considered intrusive. As the first two parameters are increased, the system resources required of the threshold detection increases. The third parameter, the threshold level, depends on the relevance of the security event being monitored and directly affects the number of false positives and false negatives reported by the system. As the threshold is lowered, the probability of false positives increase and false negatives decrease. As the threshold is raised, the converse occurs and the probability of false positives decrease and the false negatives increase.

3.1.2 Profile-based Detection

Profile-based detection is based on establishing patterns of normal behavior for a user or system and then classifying significantly deviant behavior as intrusive. It differs from threshold detection in that it employs patterns of usage instead of summary statistics to determine if an intrusion has occurred. The patterns maintained by the IDS are adaptive in that they change over time to reflect the usage patterns of each user accurately.

Profile-based detection offers a number of distinct advantages. Profile-based intrusion systems require no prior knowledge of the user to detect intrusions. The system will adapt over time to reflect the usage patterns of the account holder. An intruder who compromises the account would be detectable unless they mimic the account holder's usage patterns. Profiles also provide an easily understood summary of activity that system administrators can quickly examine and reach decisions on intrusive behavior. Finally, compared to audit records, profiles significantly reduce the amount of storage space required for maintaining security records from which security decisions can be

made [7]. The principal disadvantage of profile-based detection is that it offers no protection against insider attacks as the user can gradually train the system to accept intrusive behavior as normal [6].

3.1.3 Keystroke Monitoring

Keystroke monitoring is a misuse detection technique that monitors sequences of keystrokes for attack patterns. This is a very simplistic technique that can be easily evaded through the use of user-defined aliases or the running of intrusive programs that require non-intrusive keystroke entries [8]. While this technique was used in earlier systems, it is seldom user in modern IDS.

3.2 Artificial Intelligence Techniques

Artificial intelligence techniques are the most commonly used techniques for classifying intrusive behavior. It is also one of the earliest forms of intrusion detection and has been used in almost every IDS. There are four principal artificial intelligence techniques used for intrusion detection: expert systems, predictive patterns, neural networks, and machine learning.

3.2.1 Expert Systems

Expert systems have been and continue to be the most popular intrusion detection technique employed. Expert systems use rules in anomaly or misuse systems to detect attacks. In anomaly detection systems, the rules specify usage patterns based on selected user metrics. In misuse detection systems, the rules specify specific types of known attacks. Expert system rules are typically implemented as a series of if-then statements.

The principal advantage of expert systems is the separation of control reasoning (is this an attack?) from the formulation of the solution to the problem (system response to the attack). The disadvantage of expert systems is that they require a great deal of initial training and high maintenance during their lifetime. An expert must generate the initial rule-base, which is time-intensive and expensive. Because not every expert knows every vulnerability in a system, there is the very real chance that the initial configuration does not capture all possible vulnerabilities. As new attacks are developed, the expert system must be manually updated to capture the characteristics of the new attack.

3.2.2 Predictive Patterns

Predictive pattern-based detection is an anomaly detection technique that attempts to predict future events based on events that have already occurred [9]. Event sequences are represented as a statistically weighted set of rules based on the user profile. If user actions match n-1 events and the n^{th} event is statistically anomalous, then the system reports an intrusion. Predictive pattern systems constantly update user profiles and prune the rule set to maintain high quality patterns of user activity.

This approach has a number of advantages. Rule-based sequential patterns can detect anomalous behavior that is difficult to detect with other methods. Predictive pattern matching is also highly adaptive to changes in user behavior. This adaptively allows the system to constantly refine its rule set so that low quality patterns are continually eliminated leaving high quality patterns behind. Finally, it is easier to detect users who try to train the system during its learning phase [8].

3.2.3 Neural Networks

Neural networks are an anomaly detection technique that trains a neural network to predict a user's actions given a window of n previous actions. The network is trained through a user profile of representative user commands. If the users actions are significantly deviant from the user profile as maintained by the neural network, the system reports an intrusion [10].

Neural networks have a number of advantages and disadvantages. They cope with noisy data such as command sequences well and are not dependent on any statistical assumptions about the user. They are also easy to modify for new users. The disadvantage of neural networks is that a small event window will result in false positives while a large event window will increase the probability of false negatives. If intruders have access to an account during the learning phase, they can train the network to accept intrusive behavior as normal. Finally, the network topology is only determined after considerable trial and error [8].

3.2.4 Machine Learning

Machine learning is an anomaly detection technique that compares the user-input stream with a historical library of user commands to detect anomalous behavior. In one approach, the input stream is broken into fixed length sequences (normally 8-12 command tokens), which are compared through a sliding window against a library of 500-2000 user sequences. The library is unique for each user. The result of the comparison is a similarity measure. If the similarity measure is greater than threshold level, then the user activity is characterized as abnormal; otherwise, user activity is classified as normal [11].

The selection of several parameters greatly influences the effectiveness a machine learning system. The optimal sequence length appears to be 8-12 command tokens. Shorter sequences provide low detection rates while longer sequences increase the false positive rate and provide lower intrusion detection rates. The sliding window size determines the shortest interval in which the system can detect an intruder. Experimental results also suggest that: the ideal library size is user dependent; as the size of the library increases, the number of false positives also increases; and, the method of pruning the library significantly impacts on the effectiveness of the overall system [11].

Machine learning as an intrusion detection technique has a number of advantages and disadvantages. Machine learning does not require the selection of measurement metrics, which remains an open research issue. Instead, it measures all user actions and builds a user profile from the metrics most pertinent to each user. This flexibility in metric selection comes at a significant cost. Machine learning is computationally intensive and its effectiveness is dependent on differences between users.

3.3 Graph-based Techniques

Graph-based techniques are misuse systems that represent user and system behavior as a set of graphs that are then compared to attack signature graphs to detect intrusions. This is a relatively intrusion detection technique and has been used in a limited number of IDS. There are three graph-based techniques used for intrusion detection: state transition analysis; pattern matching, and model-based detection.

3.3.1 State Transition Analysis

State transition detection is a misuse detection technique that models a host as a state transition diagram. It was used as the basis for the USTAT system [12]. Known attack patterns are encoded as states in the diagram with the final state in a chain being the *compromised* state. The preceding states are known as *guard* states. The guard states act as a filter to separate normal from intrusive activities.

State transition detection has a number of advantages and disadvantages. Because it maintains system state over multiple user sessions, it can detect co-operative attacks as well as attacks that span across multiple sessions. It can also foresee imminent compromise states and take pre-emptive measures to prevent the system from entering a compromised state. State transition systems are limited in that the attack patterns can only specify a sequence of events rather than more complex forms. This severely limits the types of attacks that the system can detect [8].

3.3.2 Pattern Matching

Pattern matching detection is a misuse system that represents known attack signatures as patterns that are compared against audit records. Knowledge about attacks is represented as a set of specialized graphs. The graphs represent the transition from normal system states to compromised states and are an adaptation of colored Petri nets. This technique is similar to the state transition technique, but pattern matching associates guards with transitions, rather than with states. This technique has been implemented in the Intrusion Detection In Our Time (IDIOT) system in which pattern matching is used as the basis for a generic misuse detection model [8, 13].

Pattern matching has similar advantages and disadvantages as the state transition model with the following additions. Pattern matching can detect some attack signatures that the state transition model cannot and priorities can be assigned to signatures, which can be used for prioritized evaluation of attack patterns and response to intrusions. Additionally, patterns can be dynamically added to the system while maintaining the partial matches already present in the system. Pattern matching requires substantial overhead to track partial attacks that may be by different users and distributed in long periods of time. The complexity of the model grows exponentially with respect to the size of the colored Petri net as the complexity of the attack signature increases. This limits the ability of pattern matching systems to respond in real-time to complex attacks [13].

3.3.3 Model-based Detection

Model-based detection is a misuse detection technique that detects attacks through observable activities that infer an attack signature. Model-based detection has three components: an *Anticipator*, *Planner*, and *Interpreter*. The *Anticipator* uses two types of models, activity models and scenario models, to predict the next expected step in an attack scenario. Activity models are representations of current activity while scenario models represent intrusion signature specifications. The *Planner* takes the *Anticipator's* prediction as a hypothesis and translates it into audit log format. The Interpreter then uses these predicted audit entries as search strings in the audit records. If the model-based detection system accumulates sufficient evidence of an intrusion by crossing a system-defined threshold, the system reports an intrusion attempt [14].

Model-based detection has a number of advantages and disadvantages. Model-based intrusion detection is based on a mathematically sound theory of reasoning in the presence of uncertainty. Because the *Planner* and *Interpreter* are looking for very specific audit records, they can filter large amounts of the audit files, which leads to excellent performance. In addition, because the model is predictive, the system can take appropriate countermeasures to thwart the intruder's attacks. Unfortunately, model-based detection requires easily recognizable, distinguishing patterns of misuse. If the intruder can disguise their attack, this technique can be easily bypassed [8].

3.4 Information Retrieval Techniques

Information retrieval, as used in intrusion detection, is a misuse detection technique that searches for attack patterns by building an index of audit logs and then searching this index. To be used in a real-time system, the information retrieval system must maintain the audit index by periodically rebuilding the index as new audit records are generated. There are a variety of techniques for building, searching, and storing indexes that result in different tradeoffs in terms of false positives and negatives.

The use of information retrieval techniques for intrusion detection has a number of advantages and disadvantages. Information retrieval techniques have a number of techniques for finding information in a large amount of data that have been actively researched for the last forty years. These techniques have a variety of approaches and techniques for processing inexact and partial matches [15]. Index retrieval is both fast and the index files require less secondary storage than the original audit files. However, like other pattern matching techniques, information retrieval is easy to defeat by aliasing commands so that they so that the signatures of misuse are masked. Additionally, the building of the index is a processor and memory intensive technique that normally cannot be done in real-time.

3.5 Positive Behavior-Based Detection

Positive behavior-based intrusion detection is a specification-based technique that specifies intended system behavior and reports activity outside of intended this behavior. This is one of the newest approaches to intrusion detection. There are two forms of positive behavior-based systems used for intrusion detection: specification-based and transaction-based detection.

3.5.1 Specification-Based Detection

Specification-based detection uses a program behavior grammar to enunciate intended behavior and then scans audit files for violations of this expected behavior. For example, the finger daemon should only execute the finger program and should only read a very limited subset of files that can be easily specified. If the finger daemon attempts to read the system password file, this violates program specification and an intrusion would be reported [16].

This technique has a number of advantages and disadvantages. The program behavior grammar describes the behavior of security-critical programs only and only in terms of sequences of operations. It does not consider parameter value, which can be used for buffer overflow and other types of attacks. The specification of security-critical programs is subject to errors of omission and does not address those programs that

require access to security critical files. This specification process is the main limitation of this technique. On the other hand, specification-based detection can detect previously unknown attacks without reprogramming and for many types of attacks, is a natural mechanism for explicitly stating and enforcing security policy.

3.5.2 Transaction-Based Detection

Transaction-based detection is a specification detection technique that delineates allowed actions and sequences of actions through transaction management. User activity is modeled as a series of read and writes operations. The transaction-based detection system checks to ensure that all transactions are:

- Atomic (all operations are completed).
- Consistent (system remains in a consistent state.
- Isolated (transactions do not interfere with other transactions)
- Durable (transaction results are saved in permanent storage) [17].

By enforcing these four properties, a large subset of intrusive behavior can be detected. The main limitation of the transaction-based detection is the specification process. Specifying allowed transactions is time-consuming and subject to specification and management errors.

As with the intrusion detection approaches, there is no one technique that provides complete security. As such, most modern IDS employ two or more techniques to detect intrusions.

4. A COMPARATIVE ANALYSIS OF INTRUSION DECTECTION SYSTEMS

In executing the approaches and techniques discussed in Sections 2 and 3, intrusion detection is implemented as an overlay of two separate and different technologies: Network IDS (NIDS) and Host-based IDS (HIDS) systems. The primary advantage of NIDS is that it can watch the whole network or any subsets of the network from one location. Therefore, NIDS can detect probes, scans, and malicious and anomalous activity across the whole network. These systems can also serve to identify general traffic patterns for a network as well as aid in troubleshooting network problems. When enlisting auto-response mechanisms, NIDS can protect independent hosts or the whole network from intruders. NIDS does, however, have several inherent weaknesses. These weaknesses are its susceptibility to generate false alarms, as well as its inability to detect certain attacks called false negatives. NIDS also is not able to understand host specific processes or protect from unauthorized physical access. HIDS technology overcomes many of these problems. However, HIDS technology does not have the benefits of watching the whole network to identify patterns like NIDS does. A recommended combination of host and network intrusion detection systems, in which a NIDS is placed at the network border and an HIDS is deployed on critical servers such as

databases, Web services and essential file servers, is the best way to significantly reduce risk.

Generally, the commercially available IDS products shown in Table 2 are classified according to their approach to intrusion detection with all being either host or network-based. None of the products integrate host-based and network-based intrusion detection capabilities and a few integrate security assessment capabilities with basic IDS functionality, such as audit trail analysis and malicious software protection.

Vendor	Product Name	Approach to ID	Platform
ISS	RealSecure	Network-based Packet capture, signature analysis, and real-time playback.	UNIX and NT
Cisco (formerly WheelGroup)	NetRanger	Network-based. Passive network monitor with packet filtering router.	UNIX
Security Dynamics (formerly Intrusion Detection)	Kane Security Monitor	Host-based. Passive ID capabilities with assessment functions.	NT
DMW Worldwide	HostCHECK	Host-based. Passive ID capabilities (audit trail analysis and file checksums) and assessment functions.	UNIX
MEMCO (formerly AbirNet Ltd.)	SessionWall	Network-based. Real-time connection and playback.	NT

Table 2: IDS products

A description and comparison of the different systems are presented next in Table 3. To this end, some systems will receive a more elaborate description while others will just be mentioned for the passing.

Name	Description	Features, Pros and Cons
Autonomous Agents For Intrusion Detection (AAFID)	The AAFID architecture [Figure 3] has three components: *agents, transceivers*, and *monitors*. *Agents* are independent software units that monitor a limited number of aspects of a host. A host can have a number of agents, each monitoring different aspects of the host. Agents do not have the authority to generate directly an alarm or to communicate directly with each other but instead communicate through a transceiver. A *transceiver* coordinates the activities of host agents. There is one transceiver per host. The transceiver starts and stops agents as required, monitors agents, responds to monitor commands, receives and processes agent reports, and distributes information to agents or *Monitors* as required. *Monitors* perform the same roles as transceivers but control several hosts as opposed to transceivers, which control a single host but multiple agents [18, 19].	There have been two prototypes implemented using the AAFID architecture. The first prototype was implemented using Perl, Tcl/Tk, and C and was a proof of concept. The second prototype was written in Perl and is being used to test the architecture for ease of use, configurability, and extensibility. AAFID is novel in a number of ways. The use of agents provides IDS that is scalable, resilient to subversion, and provides graceful degradation of service. Losing one or more agents does not result in the loss of the entire system but instead the IDS continue to operate at reduced efficiency. Agents scale to larger systems with additional monitors providing a hierarchy of agents to detect intrusions.
Adaptive Hierarchical	AHRAB [Figure 4] is based on an adaptive, hierarchical collection of cooperating agents that	AHRAB provides graduated, risk-based intrusion detection. Unlike other systems,

Agent-Based Intrusion Detection System (AHRAB)	collectively work to detection intrusions. There are three types of software agents: *worker agents, managers*, and *directors*. *Worker agents* take the output of standard stand-alone security tools, analyze the output, and provide aggregated results to managers. *Managers* provide guidance to and adaptively control worker agents based on perceived risk and resource constraints. They aggregate the output of multiple worker agents and make decisions to adapt the security of the system. This adaptation may be: starting additional worker agents; running existing worker agents under a more robust configuration; changing the resource constraints under which the worker agents run; or adapting themselves to use a more robust reasoning mechanism. Mangers may be host-based or network-based. *Directors* provide guidance to managers and integrate the results of traffic and component-based managers to provide a comprehensive view of the network and devices functioning under the AHRAB system.	AHRAB does not provide a single level of intrusion detection. Instead, it increases or decreases system intrusion detection efforts based on the current situation. If there are indications the protected system is under attack, it will increase the intrusion detection efforts. If the system does not appear to be under attack, it will gradually reduce intrusion detection efforts until it reaches a base level set by the system security manager. The increase or decrease of intrusion detection is resource-constrained so that the intrusion detection effort is related to the probability of an intrusion. AHRAB also incorporates human feedback into its adaptive architecture. As the system detects or does not detect an intrusion, a human provides feedback to the AHRAB system. AHRAB then adjusts the creditability of the agents used for intrusion detection.
Cooperating Security Managers (CSM)	Cooperating Security Managers (CSM) [Figure 5] is a host and network-based detection system based on cooperating agents that proactively respond to intrusions without using a centralized director. Key to this approach is that there are no centralized mangers, and a proactive instead of reactive response to intrusions is used. With no centralized managers, CSMs coordinate among themselves to detect intrusions. In a proactive response environment, CSMs not only detect intrusions on their monitored hosts, but also notify other hosts if they suspect that one of their users is attempting to attack another host. Having CSMs on all or most of the host machines on a network is key to this proactive approach [20-22].	If an intrusion is detected by the *Local IDS* or *Security Manager*, the *Intruder Handler* reacts to the intrusion by taking a preprogrammed reaction. At a minimum, the system administrator is notified. Depending on the intrusion, the intrusive session may perform a number of actions including terminating the current session or locking the user's account. Finally, the *User Interface* provides the capability for the system administrator to query the *Security Manager* on the current security status
Distributed Intrusion Detection System (DIDS)	The Distributed Intrusion Detection System (DIDS) [Figure 6] is a *host and network-based anomaly and misuse detection system* that is based on the host-based anomaly and misuse IDS and the NSM system. DIDS was designed to detect a number of additional attacks that NSM had difficulty detecting through user tracing. These attacks included low-frequency doorknob and network browsing attacks [23]. During a low frequency doorknob attack, the intruder attacks a number of computers using a limited number of common account and password combinations. Because the attacker uses only a few combinations, the IDS may not detect the failed logins as intrusive. Network browsing attacks are detected similarly. During a low frequency browsing attack, users scan a number of files on several systems within a short period of time looking for vulnerabilities. The activity on any single host is not anomalous enough	DIDS addressed several shortcomings found in NSM. Unlike NSM, DIDS is able to monitor users that connect to a system through the console or dial-up lines. It is also able to perform limited user tracing even if the data traffic if encrypted. DIDS assigned a unique Network-user Identification (NID) to all users and is able to track users as they traverse the network through monitored hosts. This prevents attackers from hiding their true identity and origin by switching accounts as they log into different host computers. DIDS is able to trace users across multiple hosts by treating the network connection between users and hosts as a shared resource and examining who is accessing that resource.

	for the IDS to flag the activity as intrusive. Because DIDS can trace a single user's activity across multiple systems, DIDS can detect the intrusive behavior while other systems would have difficulty with these types of low frequency attacks [23-24]	
Event Monitoring Enabling Responses to Anomalous Live Disturbances (EMERALD)	Event Monitoring Enabling Responses to Anomalous Live Disturbances (EMERALD) [Figure 7] is a highly distributed *anomaly and misuse detection system* that employs signature analysis with statistical profiling. EMERALD is built around the concept of hierarchical, adaptive monitors that provide intrusion detection for thousands of users connected in a federation of independent domains. Each monitor may consist of up to four components depending on their role: a target specific resource object, a profiler engine, a signature engine, and a resolver. The target specific resource object contains the target specific configuration data and methods so that the monitor can remain independent of the analysis target to which it is deployed. This separation of the generic monitor code-base from the target specific code and data makes EMERALD an extensible system. The profiler engine performs statistical profile-based anomaly detection while the signature engine performs rule-based misuse detection from the event stream being monitored. The resolver is an expert system that coordinates the analysis reports from the profile and signature engines. It may incorporate results from other analysis engines outside to the monitor, and it also implements the response policy of the monitor. As intrusive behavior is detected, the resolver can employ countermeasures to limit the damage of the intrusive behavior or to provide more detailed monitoring.	There are three types of monitors that populate the EMERALD hierarchy: Service, Domain and Enterprise monitors. *Service Monitors* are dynamically deployed within a domain and provide localized real-time analysis. This analysis may be of network infrastructure components such as routers or gateways or may be networked privileged subsystems. The analysis may be passive where audit logs are read and analyzed or involve active probing of the system for additional indications of intrusive behavior. *Domain Monitors* oversee a domain and correlate intrusion reports from service monitors to detect intrusive behavior across an entire domain. *Domain Monitors* also interface with other monitors outside of the domain and report domain threats to system administrators. *Enterprise Monitors* correlate intrusion reports across multiple domains to provide analysis across the entire enterprise.
Graph-based Intrusion Detection System (GrIDS)	GrIDS is an intrusion detection system designed to detect large-scale automated attacks on networked systems. GrIDS [Figure 8] collects data on networks and hosts. It automatically generates activity graphs based on network connections and uses these graphs as signatures for automated attacks on systems. As these graphs are constructed, they have attributes that provide the necessary data to detect intrusions. Graphs are segmented into different "graph spaces" based on the type of network abuse. These different graph spaces have different latencies associated with them depending on the latency associated with a type of attack [25].	GrIDS uses a threshold-based detection mechanism. As the activity graphs are built, detection heuristics are applied and the graphs are compared against attack signatures. Intrusion detection occurs when the graph exceeds a user-specified similarity threshold.

Intrusion Detection Expert System (IDES)	IDES is a *host-based anomaly and misuse detection system* developed by SRI International in 1985. It was one of the first IDS developed and employs user profiles and a rule-based system to detect intrusions. The user profile is constructed from twenty-five user metrics. This profile is updated daily and weighted so that the most recent activity has more weight than older user activity. In addition to user metrics, IDES monitors six remote host metrics and five overall target metrics. In measuring these metrics, IDES differentiates between discrete and continuous value measures. Discrete measures are metrics that have a finite range of values and describer user or system behavior. Continuous value measures are a function of observed behavior such that the function value changes over time [26, 27].	IDES combines the output of the anomaly detection and expert system to detect intrusions. As the Receiver receives audit records, they are placed in an Audit Data Database where they are examined by both the anomaly detection system and the expert system.
Internetwork Security Monitor (ISM)	The Internetwork Security Monitor (ISM) is a *network-based misuse detection system*. ISM is a hierarchical architecture, which consists of three components: ISMs, Security Domain Name Servers (SDNS), and security workbenches [Figure 9]. The ISMs work together to combine thumbprint data connections into logical connections. SDNS provide a mechanism for ISMs to locate other ISMs over the Internet so as to exchange thumbprint information. Finally, the security workbenches provide the ability for system administrators to examine ISM results, exchange information with other system administrators, and administer security packages such as COPS [29] and SPI.	ISM extends the DIDS and NSM systems to provide user accountability and support arbitrarily large networks. While DIDS can provide user tracing across a network, it loses this tracing ability if the user passes through an unmonitored host. ISM overcomes this shortcoming through a *thumbprinting* technique. Thumbprinting assigns a signature to a data connection, based on the data flow through that connection over a specified period of time. By correlating different connection thumbprints, it is possible to detect the same logical connection from a set of different physical connections and thus trace user activity through both monitored and unmonitored hosts [28].
Multics Intrusion Detection and Alerting System (MIDAS)	MIDAS is an IDS based on rule-based, *anomaly detection*. It is used on the National Computer Security Center's DockMaster computer. The components of MIDAS are listed in Figure 10. MIDAS runs on two machines, a Multics system and a Symbolic Lisp machine. On the Multics system, the *Preprocessor* screens audit records and extracts pertinent data and transforms it into assertions for the Symbolic machine. The *Command Monitor* on the Multics system captures related security data not present in the audit records and sends it to the *Preprocessor* for transformation into assertions. The assertions are sent to the *Fact Base* through the *Network Interface*. The assertions may cause a binding of the assertion to a rule or a series of rules.	The *Statistical Database* contains both user and system statistics that characterize what the system considers normal user activity and normal system states [30].
Network Anomaly Detection and Intrusion Reporter (NADIR)	The Network Anomaly Detection and Intrusion Reporter (NADIR) is a *profile-based anomaly and misuse detection system*. It was developed at the Los Alamos National Laboratory for use on the Integrated Computing Network. NADIR [Figure 11] periodically copies audit records from host computers to the NADIR system where it examines	In applying these rules, it maintains a level of interest metric on users, which provides an overall measurement of user behavior. A high level of interest is indicative of suspicious behavior that warrants future investigation by system administrators. NADIR provides weekly reports that highlight the most suspicious users

	audit data and generates weekly user and network profiles. An expert system compares the audit data against the profiles to detect security-related activities. The expert system also looks for attack signatures among user activity and highlights questionable activity.	as well as an overview of overall network traffic [7, 31].
Network Security Monitor (NSM)	Network Security Monitor is a *network-based anomaly and misuse* IDS that uses network traffic, not audit logs, to detect intrusions. To detect intrusions, NSM reconstructs the activities of individual users from network traffic. NSM accomplished this through a variety of techniques in different versions. In its first version, NSM used a four-dimensional matrix to measure network traffic and detect anomalous traffic. This Access Control Matrix mapped source addresses, destination addresses, services and connection IDs. Each cell within the matrix contained two values: the number of packets passed through a connection in a time interval and the amount of data passed through the connection. This matrix modeling the network is compared against matrixes that model "normal" behavior" for the hosts involved and anomalies are reported. A probabilistic distribution is used to determine what is considered anomalous.	NSM was the first system to focus on network traffic and not audit logs to detect intrusion. Network-based detection offers a number of distinct advantages. Because NSM uses standard network protocols, it can monitor heterogeneous hosts running different operating systems transparently. This transparent monitoring eliminates the need to examine and transfer audit logs, which are often a high priority target for attackers. Network-based detection also eliminates the overhead associated with running IDS on a number of hosts. Instead, the cost of running the IDS is contained to the systems running NSM. Finally, NSM found that most hosts communicate almost exclusively with a very small subset of hosts using the same services. This communications signature provides an inexpensive means of identifying many intrusions. The attacker would have to mimic this communications signature to be undetected. NSM monitored activity on an Ethernet LAN [28].

Table 3: IDS systems description

5. CONCLUSIONS

This paper has examined real-time intrusion detection systems by examining IDS approaches, techniques, and systems. As the threat and reward associated with intrusions continues to increase, research in intrusion detection is closing the gap between the intrusion detection tools and hacker attack tools. While there will always be a gap between the two, progress in intrusion detection is narrowing this gap.

REFERENCES

[1] http://itmanagement.earthweb.com/columns/article.php/1025311

[2] T. Aslam, "A Taxonomy of Security Faults in the Unix Operating System," M.S. Thesis, Purdue University, West Lafayette, IN, 1995.

[3] D. E. Denning, "An Intrusion-Detection Model," *IEEE Transactions on Software Engineering*, vol. 13 (2), 1987, pp. 222-232.

[4] F. B. Cohen, "50 Ways to Defeat Your Intrusion Detection System," *Network Security Journal*, vol. 3 (12), 1997, pp. 11-13.

[5] M. Bishop, S. Cheung, and C. Wee, "The Threat from the Net," *IEEE Spectrum*, vol. 34 (8), 1997, pp. 56-63.

[6] M. Esmaili, R. Safavi-Naini, and J. Pieprzyk, "Computer Intrusion Detection: A Comparative Survey," Tech. *Rep. 95-07*, Center for Computer Security Research, University of Wollongong, May 1995.

[7] J. G. Hochberg, K. A. Jackson, C. A. Stallings, J. F. McClary, D. DuBois, and J. R. Ford, "NADIR: An automated system for detecting network intrusion and misuse," *Computers & Security*, vol. 12 (3), 1993, pp. 235-248.
[8] A. Sundaram, "An Introduction to Intrusion Detection," *Crossroads: The ACM Student Magazine*, vol. 2 (4), 1996, pp. 26-41.
[9] H. S. Teng, K. Chen, and S. C. -Y. Lu, "Adaptive Real-time Anomaly Detection Using Inductively Generated Sequential Patterns," *Proc. IEEE Symp. on Research in Security and Privacy*, May 7-9, 1990, pp. 278-284.
[10] T. F. Lunt, "A Survey of Intrusion Detection Techniques," *Computers & Security*, vol. 12 (4), 1993, pp. 405-418.
[11] T. Lane, "An Application of Machine Learning to Anomaly Detection," Tech. *Rep. 97-03*, COAST Laboratory, Department of Computer Science, Purdue University, February 1997.
[12] K. Ilgun, "USTAT: A real-time intrusion detection system for UNIX," *Proc. IEEE Symp. on Research in Security and Privacy*, Oakland, CA, May 24-26, 1990, pp. 16-28.
[13] S. Kumar and E. H. Spafford, "A Pattern Matching Model for Misuse Intrusion Detection," *Proc. 17th National Computer Security Conf.*, Baltimore, MD, October 11-14, 1994, pp. 11-21.
[14] T. D. Garvey and T. F. Lunt, "Model based Intrusion Detection," *Proc. 14th National Computer Security Conf.*, Washington, DC, October 1-4, 1991, pp. 372-385.
[15] R. Anderson and A. Khattak, "The Use of Information Retrieval Techniques for Intrusion Detection," *Proc. of First International Workshop on the Recent Advances in Intrusion Detection*, Louvain-la-Neuve, Belgium, September 14-16, 1998, Available at http://www.zurich.ibm.com/pub/Other/RAID/Prog_RAID98/Talks.html#Anderson_33, accessed on 14 April 1999.
[16] C. Ko, M. Ruschitzka, and K. N. Levitt, "Execution Monitoring of Security-Critical Programs in Distributed Systems: A Specification-based Approach," *Proc. of IEEE Symposium on Security and Privacy Conf.*, May 4-7, 1997, pp. 175 - 187.
[17] R. Buschkes, M. Borning, and D. Kesdogan, "Transaction-based Anomaly Detection," *Proc of Workshop on Intrusion Detection and Network Monitoring*, Santa Clara, CA, April 9-12, 1999.
[18] M. Crosbie and E. H. Spafford, "Active Defense of a Computer System using Autonomous Agents," *Tech. Rep. 95-008*, COAST Laboratory, Computer Science Dept., Purdue University, February 1995.
[19] J. Balasubramaniyan, J. O. Garcia-Fernandez, D. Isacoff, E. H. Spafford, and D. M. Zamboni, "An Architecture for Intrusion Detection using Autonomous Agents," Tech. *Rep. 98-05*, COAST Laboratory, Department of Computer Science, Purdue University, May 1998.
[20] G. B. White, E. A. Fisch, and U. W. Pooch, "Cooperating Security Managers: A Peer-based Intrusion Detection System," *IEEE Network*, vol. 10 (1), 1996, pp. 20-23.
[21] G. B. White and U. W. Pooch, "Cooperating Security Monitors: Distributed Intrusion Detection Systems," *Computers and Security*, vol. 15 (5), 1996, pp. 441-450.
[22] G. B. White, E. A. Fisch, and U. W. Pooch, *Computer System and Network Security*, New York, NY: CRC Press, 1996.
[23] S. R. Snapp, J. Brentano, G. V. Dias, T. L. Goan, L. T. Heberlein, C. -L. Ho, K. N. Levitt, B. Mukherjee, S. E. Smaha, T. Grance, D. M. Teal, and D. Mansur, "DIDS (Distributed Intrusion Detection System) - Motivation, Architecture, and an Early Prototype," *Proc. 14th National Computer Security Conf.*, Washington, DC, October 1991, pp. 167-176.
[24] S. R. Snapp, S. E. Smaha, D. M. Teal, and T. Grance, "The DIDS (Distributed Intrusion Detection System) Prototype," *Proc. USENIX 1992 Technical Conf.*, San Antonio, TX, June 1992, pp. 100-108.
[25] S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. N. Levitt, J. Rowe, S. Staniford-Chen, R. Yip, and D. Zerkle, "The Design of GrIDS: A Graph-Based Intrusion Detection System," Tech. *Rep. CSE-99-2*, Department of Computer Science, University of California, Davis, September.
[26] T. F. Lunt, R. Jagannathan, R. Lee, S. Listgarten, D. L. Edwards, P. G. Neumann, H. S. Javitz, and A. Valdes, "Development and Application of IDES: A Real-Time Intrusion-Detection Expert System," Tech. *Rep. Report*, SRI International.
[27] T. F. Lunt, A. Tamaru, F. Gilham, R. Jagannathan, C. Jalali, and P. G. Neumann, "A Real-Time Intrusion-Detection Expert System (IDES)," Tech. *Rep. SRI-CSL-92-05*, SRI International, April 1992.
[28] L. T. Heberlein, B. Mukherjee, and K. N. Levitt, "Internetwork Security Monitor: An intrusion-detection system for large-scale networks," *Proc. 15th National Computer Security Conf.*, Baltimore, MD, October 13-16, 1992, pp. 262-271.

[29] D. Farmer and E. H. Spafford, "The COPS Security Checker System," *Proc. 14th National Computer Security Conf.*, Washington, DC, October 1-4, 1991, pp. 372-385.
[30] M. Sebring, E. Shellhouse, M. Hanna, and R. Whitehurst, "Expert Systems in Intrusion Detection: A Case Study," *Proc. 11th National Computer Security Conf.*, Baltimore, MD, October 1988, pp. 74-81.
[31] K. A. Jackson, D. H. DuBois, and C. A. Stallings, "An Expert System Application For Network Intrusion Detection," *Proc. 14th National Computer Security Conf.*, Washington, DC, October 1-4, 1991, pp. 215-225.

FIGURES

Figure 3: AAFID Architecture

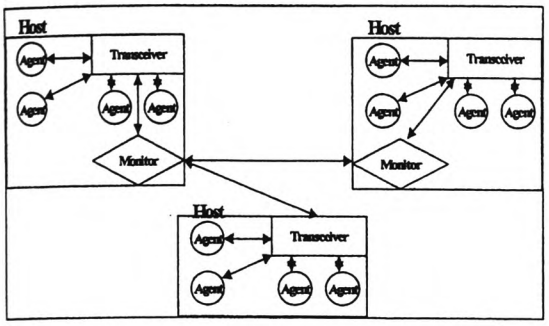


Figure 4: AHRAB Architecture

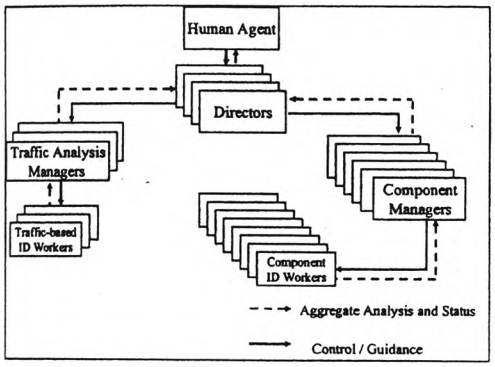


Figure 5: CSM Architecture

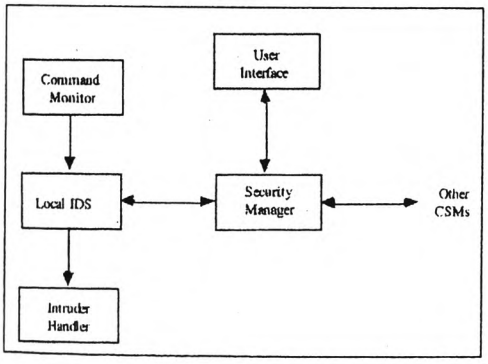


Figure 6: DIDS Architecture

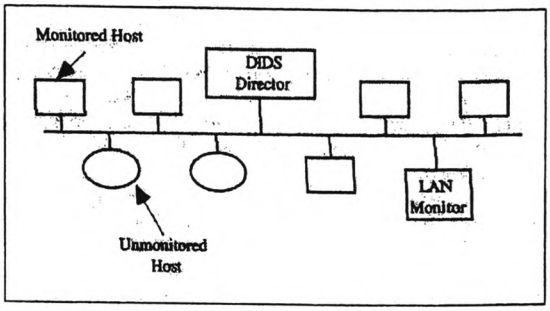


Figure 7: EMERALD Architecture

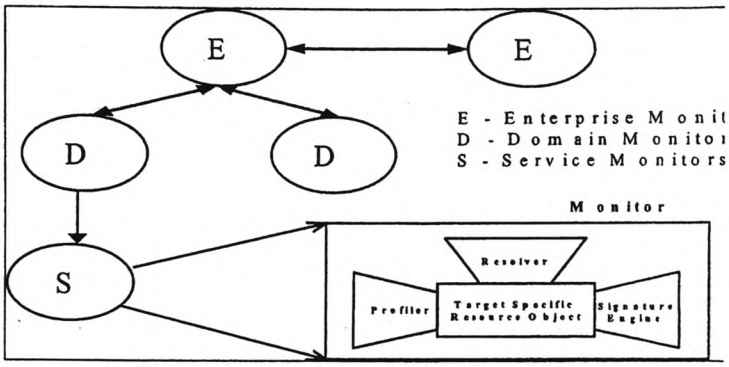


Figure 8: GrIDS Architecture

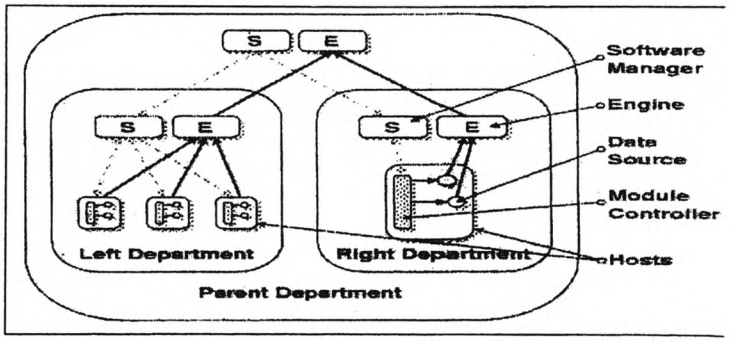


Figure 9: ISM Architecture

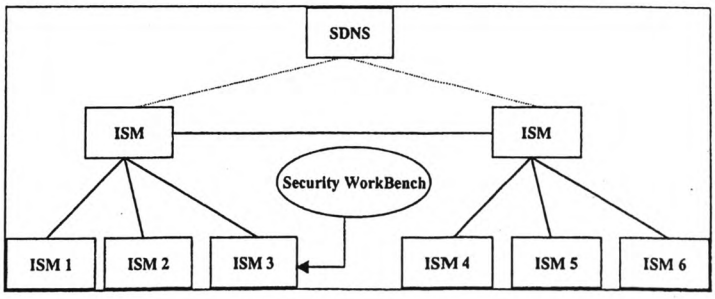


Figure 10: MIDAS Architecture

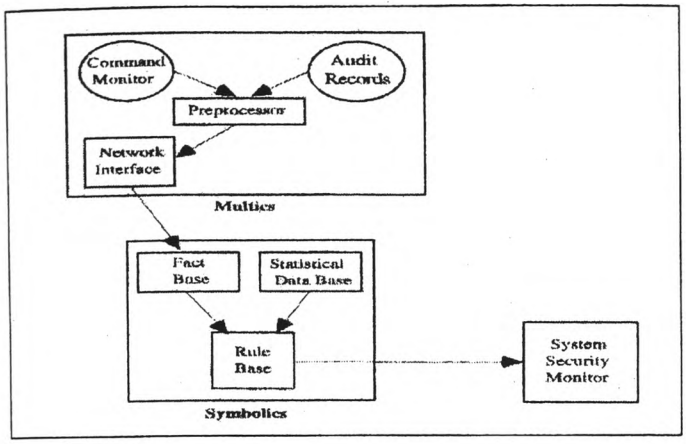
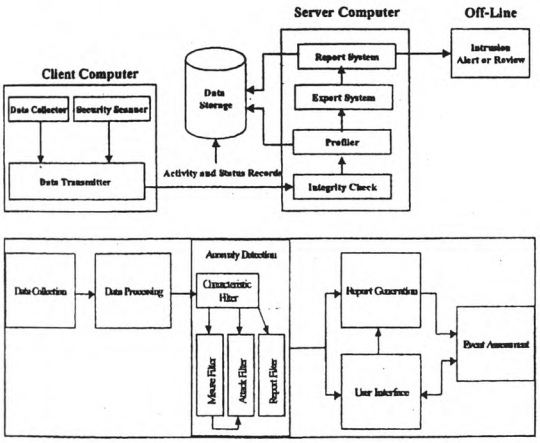


Figure 11: NADIR Architecture



Appendix 2

"Security problems and the interaction of security policies in the design and implementation of IDS in enterprise networks" In: Proceedings of the FIRST 15th Annual Computer Security Incident Handling Conference. The Westin, Ottawa Ontario, Canada, June 22-27, 2003

15TH ANNUAL

FIRST CONFERENCE

OTTAWA CANADA

June 22-27, 2003

www.first.org

inum Sponsor:

Government of Canada / Gouvernement du Canada

Office of Critical Infrastructure Protection and Emergency Preparedness / Bureau de la protection des infrastructures essentielles et de la protection civile

Binder Sponsor:

BT

Security Problems and the Interaction of Security Policies in the Design and Implementation of IDS in Enterprise Networks

By

Charles Iheagwara Andrew Blyth

Abstract

In this paper, we present the state of research and development pertinent to the design of secure systems and relate it to intrusion detection system design. We also review established standards and the application of the principles of systems engineering to network security and the methodologies used in network security design with emphasis on specifications, testing and verification. We present instantiations of the interoperation of IDS design with security policy specifications using IDS design and implementation to evaluate if and where formal methods have been used to specify, verify and validate secure IDS system properties. We also propose a conceptual design approach that relates the different phases of the design process. We then present a systematic methodology specifying validation requirements as a necessary informal method to tackle any known transport protocols issues in IDS implementation.

Key Words: Intrusion detection systems, network security design, security policies.

1.0 Introduction

Intrusion Detection (ID) systems are classified as mechanisms for parsing and filtering hostile external network traffic that could reach internal network services. These systems have become widely accepted as prerequisites for limiting the exposure of internal network assets while maintaining interconnectivity with external networks. Different ID systems have differing classifications of ``intrusion" (see Appendix 1); a system attempting to detect attacks against web servers might consider only malicious HTTP requests, while a system intended to monitor dynamic routing protocols might only consider RIP spoofing. Regardless, all ID systems share a general definition of ``intrusion" as an unauthorized usage or misuse of a computer system.

Typically, intrusions take advantage of system vulnerabilities attributed to mis-configured systems, poorly engineered software, mismanaged systems, user neglect or to basic design flaw in for instance some Internet protocols. Commercial IDS tools range from the widely available anti-viruses, to enterprise tools (e.g. Cisco/Netranger), to NT centric (e.g. Internet Security Services/RealSecure) and to configurable freeware (e.g. Network Flight Recorder).

Intrusion detection as an important component of a security system, complements other security technologies. By providing information to site administration, an IDS allows not only for the detection of attacks explicitly addressed by other security components (such as firewalls and service wrappers), but also attempts to provide notification of new attacks unforeseen by other components. Intrusion detection systems also provide forensic information that potentially allows organizations to discover the origins of an attack. In this manner, an IDS attempts to make attackers more accountable for their actions, and, to some extent, act as a deterrent to future attacks.

The design of IDS is an assembly of different components. At its most fundamental level, the IDS is a collection of detection modules also called sensors with unique attack recognition and response capabilities. Two classes are discernable:

- Network sensors: These monitor the raw, unfiltered traffic on enterprise networks, looking for patterns, protocol violations, and repeated access attempts that indicate malicious intent.

- OS Sensors: These sensors perform real-time intrusion monitoring, detection, and prevention of malicious activity by analyzing kernel-level events and host logs.

The detection modules are deployed at strategic locations across the enterprise network in order to stop attacks, misuse, and security policy violations before damage is done. When an IDS detects unauthorized activity, it can respond in a number of ways, automatically recording the date, time, source, and target of the event, recording the content of the attack, notifying the network administrator, reconfiguring a firewall or router, suspending a user account, or terminating the attack.

Because of its importance, it is critical that intrusion detection systems function flawlessly. In order to be useful, site administration needs to be able to rely on the information provided by the system; flawed systems not only provide less information, but also a dangerously false sense of security. Moreover, the forensic value of information from faulty systems is not only negated, but potentially misleading.

Due to the implications of the failure of an IDS component, it is reasonable to assume that the performance of IDS are themselves crucial to an organization's security as they could become logical targets for attacks.

The implementation of the IDS in enterprise networks has exposed the design and other pitfalls in the current implementation of commercially available intrusion detection systems. The pitfalls include the issues of variant signatures, false positives and negatives alerts, data overload, difficulties to function effectively in switched environments and scalability issues.

In this paper, we review the state of related research and development pertinent to the design of secure systems and relate it to Intrusion Detection System design and its relationship with formal methods and standards. This will include a review of established standards and the application of the principles of systems engineering to network security and the methodologies used in network security design with emphasis on specifications, testing and verification. With such a large base to draw from, some issues will obviously receive more attention than other equally as important issues.

Secondly, we present instantiations of the interoperation of IDS design with security policy specifications using IDS design and implementation to evaluate if and where formal methods have been used to specify, verify and validate secure IDS system properties. Thirdly, we propose a conceptual design approach that relates the different phases of the design process. We then devise a systematic methodology specifying validation requirements as a necessary informal method to tackle any known transport protocols issues in IDS implementation.

The paper is organized as follows. In Section 2, we discuss design standards and policies. Sections 3 and 4 discuss research and development of transport protocols and intrusion detection systems. The IDS architecture, implementation problems, and problems in routing protocols are discussed in Sections 5, 6, and 7, respectively. We define critical design concepts and requirements in Section 8 and present issues related to formal and informal specifications and verifications in Section 9. Finally, Section 10 contains conclusions.

2.0. Design Standards and the Interaction of Security Policies in Enterprise Systems

The enterprise network is a system, which interconnects a multitude of computers and devices for the purpose of communications and information/resource sharing. To keep the various interconnected parts of the system interoperable, rules and procedures must be established. In a secure processing environment, enterprise networks have additional "layers" of rules and procedures imposed, each addressing unique security requirements, with no one set of requirements (software or hardware) applicable to all security issues for any specific situation.

The design of an enterprise network is often an assembly of very dissimilar components. In enterprise networks, problems could occur because there are layers of security, each very narrowly focused for specific conditions. For emerging systems with greater capabilities and a multitude of abbreviations and operating system names, the potential exists to overlook "old" rules in favor of ever more simplistic ways of dealing with security, regardless of the layers involved. This tendency creates both the need for increased understanding of the various security layers when using shared resources in multi-secure network environments, and the need for continuing industry awareness of network security problems.

The system's security is a shared responsibility among its various subcomponents, although the ultimate burden falls on the operating system. Appropriate hardware support can minimize the impact of security features on the network performance. Hence, a network system that satisfies the enterprise Multilevel Security (MLS) policy must enforce access control: processes have access to objects in accordance with the security policy. The network system itself must also not be a channel for communication of information that violates the security policy.

The main difficulty in designing systems consisting of independent, interacting components lies in the complexity arising from often extremely large number of possible orderings in which actions of the individual components can interleave. Subtle bugs can appear in an incorrect design when events take place in a certain order.

The ultimate goal in establishing a system security standard is to insure that a level of security, consistent with security priorities, is applied to all resources throughout their procurement and deployed life cycle process. A system could be said to be secure if the information it stores is protected against release, modification, or misuse by unauthorized users.

In quality assurance, security and accessibility disciplines, System Security Engineering Management (SSEM) is used to apply systems engineering to the host of possible problems that could affect overall security requirements from concept exploration through deployment. In addition, SSEM sets the stage for long-term security control over the life cycle of the system. This also includes controlling the injection and proliferation of classified information over networks (stand-alone, trusted gateway controlled, or multi-level). Numerous standards exist which define user restrictions, equipment capabilities, network management controls, audit trails, etc.

The Orange Book describes the Multilevel Security Policy (MLS) in the context of users and objects, and requires that a user get to see the contents of objects at his or her levels but never lower levels. More abstract models of security that avoid the need to consider objects were formulated by different researchers. In these models, the information a user observes is to be dependent on the actions of users at his higher or lower level. That is, lower level users cannot observe the actions of higher-level users.

Ratings for a secure system according to the system's services in support of security and the extent of the certification of the system with respect to a security policy has been defined in the Orange Book. A1 and beyond -A1 are the highest ratings granted to systems that have been formally verified to the satisfaction of a security policy. A1 certification requires that the system design be verified, while beyond –A1 is more stringent in that it requires verification of system's implementation.

For a single host system, the design is considered to be specifications of the functional behavior of each service provided by the system, e.g., system calls and ordinary instructions accessible to user processes. Once the interface specification of a system has been verified, the implementation must also be verified. Such verification is deemed satisfactory when the executable code is verified. This approach leads to elimination of errors that could render a system insecure through the verification of design decisions that can be formulated in stages of development well before the code is produced.

In recent years, different standards specific to applications and processes have been developed. For instance, Open systems are often governed by specific applications programming interface standards (APIs). APIs address such topics as Operating Systems commands and utilities, Data Base Access (Structured Query Language or SQL), programming languages, Graphic User Interfaces (X Window

System). A significant effort has been made by the industry to define and standardize many of these system aspects, especially when applied to protected systems.

In the Open System Interconnection (OSI) standards, five distinct areas have been identified for network management systems: fault and problem management, performance management, configuration management, accounting, and security management. The integration of these functions, on a single platform, constitutes integrated network management. The network management function most important to SSEM is obviously security management.

The capability to perform classified and/or non-classified processing at will, while at the same time allowing open access to other unclassified networks in the open systems is a very difficult problem. This is because of the architecture and inter-process relationships between the various business units/processes in most open system networks. The dominant layered model for organizing communications protocols in open system networks is the one developed by the International Organization for Standardization (ISO) shown in Appendix 2. This is a seven-layer protocol model known as the Open Systems Interconnect Reference Model (OSI/RM).

Certifications are also inclusive of standard provisioning applicable to network security standards, with numerous security certification requirements imposed on individual processing and communication equipment used in networks. Because of the fast pace of emerging equipment capabilities, many of these requirements are under constant pressure to be streamlined and simplified for the ease of use by those who often don't understand all the technology issues behind integration and communication security. When those who don't understand are faced with making a decision, often the potential threat issue, regardless of technical concern, is relegated to a position of less importance. No area is this more important than open systems.

In virtual private networks (VPNs), standards have been established to secure the system. Conformance is attained by implementing a public key infrastructure (PKI) that can be either contracted through a service or implemented in-house, depending on cost, security policies, and other requirements. Cost per connection for a service is weighed against the total equipment, training, maintenance, and management costs spread over the number of connections required for an in-house VPN. Another important consideration is who will maintain control of the equipment.

A PKI starts with a certificate authority (CA), which is a software package that operates in high-security area and issues digital certificates. A certificate (cert) binds a public key value to a set of identifying information for the entity associated with the corresponding private key. The party that needs to use and rely upon the accuracy of that public key uses the cert for authentication, encryption, or digital signature. A PKI also includes a directory service for making the certificate widely available and, at a minimum, an X.509v3-compatible database for storing the certificates and information required to authenticate certificate owners.

The CA operator issues the digital certificates to the end entity—IPSec endpoints in an IPSec VPN implementation—and records the information in the database. When a certificate is compromised or is no longer correct for some reason, the CA operator lists it on a certificate revocation list (CRL). Each time an IPSec endpoint checks the validity of a certificate presented for authentication, it checks the CRL; if that certificate is listed in the CRL, it is invalid and the endpoint rejects it.

A certificate policy (CP) delineates the requirements for receiving a certificate from the CA. For example, it could require a certificate to be requested in person, along with two forms of ID. The CP also defines a level of authority, such as allowing signature authority for up to one million dollars. For an IPSec endpoint, the CP defines what information must be submitted to the CA for certification, and it should also specify what security requirements the CA must meet. To successfully implement a CA, the operator must write a certificate practice statement (CPS), which spells out how the operation of the CA matches the certificate policy requirements.

The application of standards in the design of intrusion detection systems is not clearly established judging from practical experiences with their operational performance. This could be due to difficulties in the establishment of uniform design standards that have been made worst by the ever-changing nature of traffic streams and networking technologies.

In the next section, we take a look at how formal methods have helped to develop standards in transport protocols design.

3.0. Research and Development of Standards in Transport Protocols Design

Increasingly, formal methods continue to be a suitable alternative approach to ensuring the quality and correctness of protocol designs, overcoming some of the limitations of traditional validation techniques (e.g., simulation and testing).

Research work abounds on the use of formal methods for the analysis, design and verification of transport protocols. The concern here is the correctness of the design and conformance to established security policies.

These methods have proved successful at discovering flaws, especially in the areas of correcting existing protocols, that were previously unrecognized. Desmedt et al. [1997] criticized formal verification of key distribution protocols, claimed to be secure under BAN logic, but which have already been broken. Coupled with a flaw in the basic philosophy of BAN-like logics that do not prove that a weakness in the protocol implies a violation of the basis of the crypto scheme. This has given doubts to the validity and adequacy of existing techniques on their ability to provide a proof about the correctness of a given protocol.

A fair analogy is the verification process of general-purpose computer programs, where reliable testing techniques allow many bugs to be detected, but will not provide a basis for complete proof of correctness. In the light of this, it would be a prudent and mature trend to design specific methods and implement tools, in order to aid the initial correct design of cryptographic protocols. In this case, the incorporation of formal methods into the design process can be implemented in various ways.

Firstly, Meadows C. [1995] propose that protocol design methodologies should lend themselves to or incorporate elements of formal method analysis. This is exemplified by the modular design proposed by Heintze et al. [1995]. Using protocol security reasoning tools and a composition theorem, they can state sufficient conditions for combining two secure protocols to form a new one with similar properties. Based on secret-security and time-security notions, they can provide examples of how unmet conditions result in an insecure protocol.

Secondly, Gong et al. [1995] propose that design principles could be used to develop protocols whose security is easy to evaluate. Building on an earlier work, the concept of a fail-stop processor, which, when failing, stops before any effect is visible to the outside environment and the notion of fail-stop protocols. Similarly, a fail-stop protocol halts in response to active attacks interfering with the protocol execution. The security analysis of such a protocol involves only the examination of possible passive attacks such as eavesdropping. It is therefore much easier to conclude whether the secrecy assumption can be violated.

Three phases of the proposed proof methodology for a fail-stop protocol are as follows:

- Verification that the protocol is fail-stop,
- Validation of the secrecy assumption,
- And the application of BAN-like logic.

According to Nessett [1990] the methodology applies BAN-like logic because the residue from the execution of a fail-stop protocol could be useful to an attacker. Another encouraging point for this

methodology is that the specifications of fail-stop protocols satisfy some of the main prudent engineering principles from [6]. Accordingly, if the GNY logic is used to analyze a fail-stop protocol, the proof complexity can be dramatically reduced. The research investigation shows that many existing protocols prove to be fail-stop [Gong et al 1995]; therefore the new notions are not too limiting.

Meadows C. [1995] propose a stepwise-layered methodology that can be integrated with the Heintze et al approach [1995], which is based on a stack of models at different levels of abstraction. As a first step, the protocol designer uses a relatively abstract model to construct and verify the security protocol. If this protocol is correct at that top layer, the designer focuses on a more detailed model, which refines the abstract one. The repeated execution of this process leads to the final production of a detailed specification. Much of the existing work on requirements specifications has this specific flavor. The application of BAN Logic is based on a parser that translates members of a limited class of protocol specifications into BAN Logic.

Drawing from the above framework, [Meadows C. [1995], Rudolph C. [1998] introduced an approach for designing an abstract model for cryptographic protocols that can be used as the top layer of a layered design method. The main idea is the usage of Asynchronous Product Automata. The whole design process starts with a relatively abstract model at the top layer and ends in a refined specification that can be proven to be an implementation of the top level. This model reaches a higher level of abstraction than the model presented by Heintze et al. [1995] through the use of logical secure channels, instead of encryption.

The channels technique was used by Buttyan et al. [1998] to present a simple logic for authentication protocol design. These channels are abstract views of various types of secure communication links between principals. The way channels are used is similar to the use of Pi calculus channel primitives. The proposed Simple Logic preserves the simplicity of the BAN logic and adopts some concepts from the GNY logic. It consists of a language and a small number of inference rules. The language is used to describe assumptions, events, and the protocol goals. The inference rules are used to derive new statements about the system. The goal of the analysis is to construct a witnessing deduction, which is a derivation of the goals from the assumptions and the formal protocol description. The protocol is correct in the case where such a deduction exists. The lack of a witnessing deduction means that the protocol may not be correct.

Boyd et al. [1994] propose another technique for designing key exchange protocols, which are guaranteed to be correct in the sense that a specified security criterion will not be violated if protocol principals act correctly. This technique is developed from basic cryptographic properties that can be expected to be held by a variety of cryptographic algorithms. Protocols can be developed abstractly and any particular type of algorithm that possesses the required property can then be used in a concrete implementation.

Gollmann [1996] suggest that the design of authentication protocols has proven to be error prone partly due to a language problem. The objectives of entity authentication are usually given in terms of human encounters while we actually implement message-passing protocols. The author proposed various translations of the high-level objectives into a language appropriate for communication protocols.

Several researchers believe that in the near future, more effort will be spent on designing secure protocols and less on formal verifications. Specifically, Meadows [1995] argue that design specifications do not guarantee that protocols will meet security goals that were not foreseen by the design approach, that the protocols designed are sometimes impractical, and that - due to the imprecision of design principles - flawed protocols may in any case be designed.

The existence of formal methods in the development and design of transport protocols by now is well established as a safe bet towards ensuring the efficacy of the protocols design process. At the same time, experiences in the implementation of the protocols have consistently demonstrated flaws that at times have encapsulated or at best negated the very security policies that formed the basis for their design.

Consequently, it is becoming increasingly important to complement formal methodologies with informal methods, i.e., simulation, testing and validation as a guarantee for the correctness of protocol designs. In Section 8.0 we present known issues with the implementation of transport protocols.

4.0. Research and Development of Standards in Intrusion Detection Systems Design

The development of intrusion detection systems has been studied under different contexts. The following section provides an overview of the studies.

Research into and development of automated Intrusion Detection Systems (IDS) has been under way for well over 12 years. At present a great number of systems have been deployed in the commercial or government arenas, but all are limited in what they can do. This brings to focus all the issues involved in the full cycle development of Intrusion Detection Systems.

Research and development studies are identified into three categories:

1. Modeling - Misuse or anomaly detection;
2. Analysis; and
3. Optimization techniques.

In the misuse detection model, detection is by looking for specific patterns or sequences of events representing previous intrusions (i.e., looking for the "signature" of the intrusion). It is a knowledge-based technique and only known intrusions can be detected by it. This is a more traditional ID technique, which is usually applied, for instance, in anti-virus tools.

In the anomaly detection model, intrusion detection is by detecting changes in the patterns of utilization or behavior of the system. Building a model that contains metrics derived from normal system operation and flagging as intrusive any observed metrics that have a significant statistical deviation from the model perform it. The approach is behavior-based and should be able detect previously unknown intrusions. It is in the research and development area in which currently innovative modeling paradigms are explored which are inspired from biological systems. Pioneers in this area are from the University of New Mexico whose work is based on the idea that intrusion detection systems should be designed to function like the way the human natural immune systems distinguish between "self" from "non-self" antibodies.

The main challenge with this approach, like for every behavior-based technique, is to model the "normal" behavior of a process. Learning the activity of the process in a real environment can do this. Another approach, advocated by IBM research, consists of describing the sequences of audit events (patterns) generated by typical UNIX processes. Another method developed by Nokia is based on Kohonen Self Organizing Maps (SOM).

Analytical studies of ID systems attempting to address the issue of network surveillance include the Network Security Monitor developed at University of California at Davis (UC Davis), and the Network Anomaly Detection and Intrusion Reporter developed at Los Alamos National Laboratory. Both perform broadcast LAN packet monitoring to analyze traffic patterns for known hostile or anomalous activity. Further, research by UC Davis in the Distributed Intrusion Detection System and later Graph-based Intrusion Detection System projects attempted to extend intrusion-monitoring capabilities beyond LAN analysis, to provide multi-LAN and very large-scale network coverage.

Morris [1985] investigate network traffic intensity measurement. Intensity measures distinguish whether a given volume of traffic appears consistent with historical observations. These measures reflect the intensity of the event stream (number of events per unit time) over time intervals that are tunable. Alternatively, a sharp increase in events viewed across longer durations may provide insight into a consistent effort to limit or prevent successful traffic flows. Morris investigated intensity measures of

transport-layer connection requests, such as a volume analysis of SYN-RST messages, which could indicate the occurrence of a SYN-attack against port availability (or possibly for port scanning). Maimon [1985] explored intensity measures of TCP/FIN messages as a variant considered to be a more stealthy form of port scanning.

Morris [1985] contend that monitoring overall traffic volume and bursty events by using both intensity and continuous measures provides some interesting advantages over other monitoring approaches, such as user-definable heuristic rules that specify fixed thresholds. In particular, the intensity of events over duration is relative in the sense that the term ``high volume" may reasonably be considered different at midnight than at 11:00 a.m. The notion of high bursts of events might similarly be unique to the role of the target system in the intranet (e.g., web server host versus a user workstation).

Mounji et al. [1995] analyze traffic streams using Signature Analysis techniques. Signature analysis is a process whereby an event stream is mapped against abstract representations of event sequences known to indicate the target activity of interest. Determining whether a given event sequence is indicative of an attack may be a function of the preconditions under which the event sequence is performed.

Lunt et al [1989] investigate the use of coding schemes for representing operating system penetrations through audit trail analysis. Using basic signature-analysis concepts, the authors demonstrated that some detection methods could support a variety of analyses involving packet and transport datagrams as event streams. For example, address spoofing, tunneling, source routing, SATAN attack detection, and abuse of ICMP messages (`Redirect` and `Destination` Unreachable messages) could all be encoded and detected by signature engines that guard network gateways.

Lunt et al. [1989] also investigate "Off-line" vs. "Real-time" analysis as another area where more conventional classification divides IDS's into systems which operate after the event and rely on analysis of logs and audit trails for preventive action and those that attempt real-time monitoring in the hope that precursor signs of abnormal activity give indication that corrective action is possible before a damage occurs.

Denning et al. [1987] emphasize the different aspects of session activity within host boundaries given the fact that the primary input to intrusion-detection tools, audit data, is produced by mechanisms that tend to be locally administered within a single host or domain. However, as the importance of network security has grown, so has the need to expand intrusion-detection technology to address network infrastructure and services.

Jacobson et al. [1993] investigate fault detection and diagnosis in computer network and telecommunication environments within the framework of alarm correlation. The high-volume distributed event correlation technology promoted in some projects provides an excellent foundation for building truly scalable network-aware surveillance technology for misuse. However, these efforts focus primarily on the health and status (fault detection and/or diagnosis) or performance of the target network, and do not cover the detection of intentionally abusive traffic in distributed and switched environments. Indeed, some simplifications in the fault analysis and diagnosis community do not translate well to a malicious environment for detecting intrusions. For examples, assumptions of stateless correlation, which precludes event ordering; simplistic time-out metrics for resetting the tracking of problems; ignoring individuals/sources responsible for exceptional activity.

As the scale of scientific research of IDS systems grows by leaps and bounds, so does the nature of IDS interoperation, architecture and implementation.

The advent of large scale commercial intrusion detection systems tend to have given a relative assurance to the information technology community that has been very anxious to maximize the use of these highly advertised ID systems as added armor to secure network systems. Many IDS products have been deployed in commercial and corporate networks. With this has come a shift in research focus in so many areas. One of such is in the area of the IDS performance.

IDS evaluation studies treat the relationship between deployment techniques and attack system variables and the performance of the IDS.

Richards [1999] evaluate the functional and performance capabilities of the industry's leading commercial type IDS. In the areas tested, the performance of the IDS was rated based on their distinctive features, which were characterized into different performance indexes. The research work represented a new direction for IDS in that it moved the focus away from scientific concepts research to performance evaluation of the industry's best products. However, the study was limited to a small proto design isolated and non-switched network which did not reveal the impact of packet switching on the accuracy and ability to capture attack packets in their entirety.

Iheagwara et al. [2002] investigate optimization of intrusion detection systems deployment techniques in switched and distributed systems. They demonstrated that monitoring techniques could play an important role in determining the effectiveness of the IDS in a switched and distributed network.

Porras et al. [1198] discuss IDS failures in terms of deficiencies in accuracy and completeness, where accuracy reflects the number of false positives and completeness reflects the number of false negatives. All of the above research works predated the advent of Gigabit networks. The scalability issues associated with IDS deployment in Gigabit environments have opened up a new area of research.

The problem here is that with the advent of Gigabit Ethernet, not only is there a significant increase in the bandwidth – and thus a significant increase in the volume of traffic to be analyzed – but also a move into the realms of the purely switched network. Because in the promiscuous mode, sensors can only see traffic on its own segment, and in a switched environment, every connection to the switch is effectively a single segment. In the older technologies of 10mbps or 100mbps bandwidths, this can be overcome by the use of network taps or mirroring all the switch traffic to a span port, to which the IDS sensor is attached. But with Gigabit, the result would be a seriously overloaded sensor.

Currently suggested solutions include building an IDS technology into the switch hardware itself that will allow the sensor to grab traffic directly from the backplane or in the alternative move to a pure Network Node IDS implementation where the agents are concerned only with the traffic directed at the host on which they are installed.

Using newly deployed Gigabit technologies, Iheagwara et al. [2002] explored the relationship between traffic variables and IDS performance for Gigabit environments. Further, they evaluated the performance of the IDS in the context of both Megabit and Gigabit environments.

The creativity of attackers and the ever-changing nature of the overall threat to targeted systems have contributed to the difficulty in the effective performance of currently available systems, especially in effectively identifying intrusions. While the complexities of host computers are already making intrusion detection a difficult task, the increasing prevalence of distributed networked-based systems and insecure networks such as the Internet has greatly increased the need for intrusion detection.

Based on what is known on the performance of the systems and the numerous problems, the models, policies and design principles have not been very effective not at least at the level of addressing the various security issues that earlier designs, i.e., transport protocols, were faced with in the past.

In order to properly analyze the performance issues arising from IDS design, we review the standard IDS architecture in the next section.

5.0 The Intrusion Detection Systems Standard Architecture

The current architecture of commercially available IDS products is built primarily out of the perceived role of the IDS. It is equally true that due to the complexities in evolving a uniform IDS technology, the current

implementation is far from achieving the desired goals. The present IDS architecture shown in figure 1 can be decomposed into the following components: Quantitative and Qualitative architectures.

Quantitative evaluation architecture: An IDS sensor's job is to watch the network and detect attacks, a role that is performed by the packet-processing engine. To do this, the sensor looks at every packet on the network it is watching. The busier the network, the more packets there are to watch. If the sensor can't keep up, it will start to miss (or drop) packets. In the case an attack spans multiple packets, the sensor holds the packets, assembles them and makes a determination on whether there is an attack. The extent and scope of accomplishing the above roles is the gauge of the effectiveness of the IDS and that is why the IDS performance is evaluated based on the ability of the processing engine effectively filter and reassemble packets to any given network throughput.

Equally important is the functionality of the IDS. In this case, the architecture is designed to define the operational setup that is used to assess the attack set detection, configuration alert triggering, logging and reporting facilities.

Qualitative evaluation architecture: The architecture defines the evaluation criteria of the IDS based on certain usability features such as the ease of user interface (ease of use, ease of configuration, ease of filter customization); integration and interoperability with operating systems and existing network infrastructure; product maturity; company focus and price.

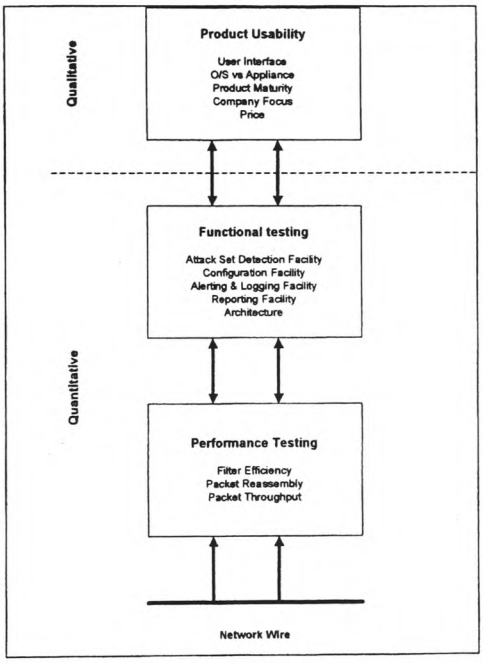


Figure 1. The standard IDS architecture

Generally speaking, implementation of the above architecture has produced varying results. On one hand, the technologies work and on the other hand, there are a myriad of problems that are largely due to the difficulty to match the current technologies with the ever-changing policies in the enterprise network. The policies are always driven by changes in the security needs that are occasioned by the rise and spur of different information system environments.

In the next section, we review the failures of the present IDS architecture that have given rise to some very serious security concerns.

6.0 Problems with the Current IDS Design

Intrusion detection as an important component of a security system, complements other security technologies. By providing information to site administration, ID system allows not only for the detection of attacks explicitly addressed by other security components (such as firewalls and service wrappers), but also attempts to provide notification of new attacks unforeseen by other components. Intrusion detection systems also provide forensic information that potentially allows organizations to discover the origins of an attack. In this manner, ID systems attempt to make attackers more accountable for their actions, and, to some extent, act as a deterrent to future attacks.

Effective implementation of IDS security facilities requires the ability of the IDS to integrate with existing network infrastructure and its interoperation with other security implementations on the protected network. At the same time, the requirements should not impose an usual burden on the IDS and thus impair its ability to be effective in capturing all traffic that originate from all specified network internally protected and Internet traffic or its compliance with specified security policy. In particular, the IDS should be able to carefully monitor those units that statistically originated most of the security attacks.

As with any other technology, there are pitfalls in the current implementation of commercially available Intrusion detection systems. The pitfalls include the issues of variant signatures, false positives and negatives alerts, data overload, difficulties to function effectively in switched environments and scalability issues.

Variants. While the ability to develop and use signatures to detect attacks is a useful and viable approach, there are shortfalls to only using this approach that should be addressed. Signatures are developed in response to new vulnerabilities or exploits that have been posted or released. Integral to the success of a signature, it must be unique enough to only alert on malicious traffic and rarely on valid network traffic. The difficulty here is that exploit code can often be easily changed. It is not uncommon for an exploit tool to be released and then have its defaults changed shortly thereafter by the hacker community.

Catch-up. New signatures can only be developed once an attack has been identified. Therefore between the creation of an attack and the deployment of a signature to detect the attack, a window of opportunity exists for an intruder to mount an attack with little to no chance of the attack being detected.

False positives. A common complaint is the amount of false positives an IDS generates. Developing unique signatures is a difficult task and often times the vendors will err on the side of alerting too often rather than not enough. This is analogous to the story of the boy who cried wolf. It is much more difficult to pick out a valid intrusion attempt if a signature also alerts regularly on valid network activity. A difficult problem that arises from this is how much can be filtered out without potentially missing an attack.

False negatives. Detecting attacks for which there are no known signatures. This leads to the other concept of false negatives where an IDS does not generate an alert when an intrusion is actually taking place. Simply put if a signature has not been written for a particular exploit, there is an extremely good chance that the IDS will not detect it.

Data overload. Another aspect, which does not relate directly to misuse detection but is extremely important is how much data can an analyst effectively and efficiently analyze. That being said the amount of data he/she needs to look at seems to be growing rapidly. Depending on the intrusion detection tools employed by a company and its size, there is the possibility for logs to reach millions of records per day.

Difficulties in switched environments. Network capture and analysis in a switched LAN environment usually means "tapping" the switch's lines by using a "mirror" port or deployment in other tapping configurations. In this approach, traffic is copied from one "source" port to another destination or "mirror" port.

It has been known that mirroring a full duplex source port may cause packet loss as traffic on the full duplex source port exceeds the available bandwidth of the mirror port.

Scaling up: In the last couple of years, there has been a significant increase in network traffic utilization. With this has come the introduction of Gigabit Ethernet technology to accommodate this increase in bandwidth – and thus the volume of traffic to be analyzed. The problem associated with this is that older IDS technologies that operate at 10mbps or 100mbps bandwidths are overwhelmed with the increase in traffic volume. With Gigabit, the older IDS technologies become seriously overloaded.

Some of the operational experiences and problems are discussed in depth as follows.

Experience has shown that the IDS performance and its stability (i.e., ability to function within design limits without failure) are determined by the following:

- Design limitations
- Traffic rate (number of packets per second)
- Traffic type (i.e. HTTP, FTP, SMB, SMTP, etc.)
- Packet size
- Session lengths
- % Of fragmentation
- Number of sessions/Hosts.
- Number of signatures active
- Workstation hardware
- Half/Full duplex transfer mode

Design limitations

Evaluation of operational results requires a methodical analysis of the many factors that could affect the IDS performance in actual network environments. This is because it is possible for the IDS to perform differently even under the same parametric specifications but different environmental contexts.

For instance, there could be cases of attaining a 100% detection rate when 100% of the traffic was scripted, but when background/normal traffic or encrypted traffic is used or added, the performance goes down. Equally, it is possible to toss 40Mbps of traffic at the IDS that won't phase it, and another 40Mbps that will phase it. In this regard, experience has been that what breaks an IDS is more often packets per second than the Layer 7 content [Iheagwara 1999], although both are relevant.

Generally, there are three bottlenecks that affect the performance of the IDS in real world environments.

- Raw sniffing speed
- Signature degradation
- Memory

Raw sniffing speed:

Sniffing speed as a measure of how much packets per seconds can be captured is a very important factor when evaluating the performance of ID systems. This is due to the fact that this could be used as a baseline when determining the maximum packet capture/second in order to quantify the operational bandwidth limits after which the performance of the IDS begins to diminish. Thus, it is a valuable measure that shows the maximum load at which the IDS will still operate effectively. The figures available from some IDS vendors as performance bottlenecks are:

- 200,000 packets/second for Cisco's Secure;
- 70,000 packets/second for Intrusion.com's Gigabit sensor; and
- 700,000 packets/second for ISS's NetworkICE Gigabit sensor.

Of interest here is NetworkIce's 700,000 packets/second sniffing rate. This means that given optimum conditions, the Gigabit sensor's engine should be able to process 700,000 packets per second. The RealSecure sensor will not sniff beyond 100,000 packets/second. It is assumed that the packets related to the above numbers are true for all (typical) packet sizes.

Consequently, what this means is that seven RealSecure sensors will be required to match the performance of NetworkICE 's sensor for a 700,000 packets/second capture in any given identical context.

In analyzing the operational results, vendor provided data should be used as the baseline reference in setting a comparison standard.

Signature degradation issue:
The second bottleneck is that NIDS (network IDS) analysis at high rates comes with signature degradation. Most NIDS use "pattern-matching routine" (signature-based), which slows/degrades with successive addition of signatures. Network ICE uses "state-based protocol-analysis", which means that it does not slow down as you add signatures because it follows a decision tree. This means that when running in the 1-Gbps ranges, all signatures can be enabled. To solve the problem of false positive alerts, filters can be set up on some signatures, thereby making it not necessary to remove signatures in order to performance tune.

The RealSecure IDS uses the pattern matching technique that somewhat impairs its functionality because the pattern matching technique degrades with an increase in the number of active signatures.

The theory behind interpreting IDS performance, by comparing "state-based protocol-analysis" vs. "pattern-matching" techniques could be explained from the perspective of the two fundamental advantages that state-based protocol-analysis has over pattern-matching in regards to the performance:

1. More efficient processing of traffic.
2. Scales better as you add more signatures.

A good example would be to compare how an IDS looks for RPC exploits. A pattern-matching system looks for patterns on ranges of ports where RPC programs typically run. For example, it might look on ports in the range 634 through 1400 for the AMD exploit. In contrast, a state-based system can remember which ports the AMD service is running on, and only test the AMD signatures on those ports that are actually running AMD. If no system on the network is running AMD, then a state-based system will never test network traffic for those signatures.

The theory behind this is that a pattern-matching system doesn't know the contents of the packets, and must match that packet for many different patterns. In contrast, a protocol-analysis system knows the contents of the packet, and only tests signatures that apply to those contents.

Given an average packet, a pattern-matching system might have to match for 10 different patterns within that packet. In contrast, on average, a state-based protocol-analysis system tests less than 0.1 signatures per packet.

This doesn't come for free: the state-based protocol-analysis that knows whether or not it should test for signatures itself costs the same as testing for a couple of signatures. Thus, the per-packet cost for pattern matching might be 10 signatures, and the per-packet cost for state-based protocol analysis might be 2 signatures.

The second part of the theory is that for pattern-matching systems, the more signatures you add to the system, the slower the system becomes. If you look in the documentation for an average sensor, it will have a comprehensive discussion on how to remove signatures in order to improve performance. This isn't applicable to a state-based protocol-analysis system.

A good example is to consider looking for Telnet login strings. There are many well-known login names that rootkits will leave behind on the system. A pattern-matching system must scan all Telnet traffic for all these patterns -- the more patterns you add, the slower it becomes.

In contrast, a protocol-analysis system will decode Telnet and extract the login name. It can then lookup the name in a binary-matching tree or a hash table. The difference is that a pattern-matching system must

match for patterns within network traffic, which scales poorly. In contrast, a protocol-analysis system pulls out a field from network traffic, and matches for that field within an internal table, which scales very well.

Again, not in the Telnet example that a username signature is only tested against the username field -- another demonstration of the first point that a packet is only tested for a signature when needed, and not when it isn't needed.

This is the theory behind the comparison. In practice, there are a lot of issues that can become more important. For example, CPU speed is doubling every year.

Given the above, the development of design standards should draw from the practical experiences. Considering the newness of the protocol – analysis technology, it will take another three to four years before its performance in the enterprise network is evaluated. And only until then will it be realistic to set a standard on which way to go.

Memory:
All currently available network intrusion detection system (NIDS) track TCP connections because they have to reassemble them, or risk being evaded. The problem here is that Gigabit networks in most cases have millions of outstanding TCP connections. This causes most boxes to fail over. For example, the architecture of the NetworkICE sensor incorporates memory- saving techniques that optimize memory consumption in preference to speed. So also does the ReakSecure architecture hold well with memory consumption.

Typical traffic
When evaluating the performance of the IDS, network throughput is important. This is commonly expressed in either Megabits (Megabytes) or Gigabits (Gigabytes). A crucial question is how many megabits (Mb) can the IDS handle before its performance nosedives?

Gauging the performance of the IDS is a function of many variables. For instance, if a packet of 1500 byte that is invalid or contains no interesting information is loaded on the network at a high rate, it will not be effective in testing the IDS. To characterize the true bandwidth limits within which the IDS is effective, the processing power of the IDS must be tested using properly configured packets. It is not just enough to send 100Mbits of 512 byte packets with a traffic generator. There is the need for a traffic that is close or identical to real traffic from real machines that is repeatable; yet still random enough that one does not end up with the vendors catering to bandwidth benchmark. That is why it is necessary to use traffic that is identical to real traffic from real machines in a performance evaluation.

Another dimension here is the variable nature of traffics on most networks. Traffic varies greatly from network to network. Internal enterprise networks might see a lot of SMB, NFS, SNA, and SQL network traffic. For example, while external/DMZ networks might see mostly HTTP, FTP, SMTP, and the occasional SSH session, a university network will see a lot of HTTP, FTP, SSH, SMTP, IMAP, POP, Napster, IRC, and a myriad of other protocols that you won't see in the average corporate space and in a carrier network will see everything from HTTP traffic to BGP updates, and every other protocol that goes across the network.

The point is that there isn't really an easy way to say "typical traffic." One might be able to craft some baseline assumptions on what university traffic looks like, what internal corporate traffic looks like, what DMZ/external corporate traffic looks like, what ISP traffic might look like, etc., but environments are so wide and varied that there is no "one size fits all" approach to traffic modeling. For example, sending 100Mbits of a typically used protocol (like HTTP) could crush an IDS that wouldn't produce the same result with for instance, 500Mbits of UDP traffic on a non-standard port.

Defining a uniform standard for all vendors will serve as a useful benchmark.

Packet size
Instances exist when the attainment of maximum (%100) utilization will have different meaning depending on the context. For instance, in analyzing an output such as the one depicted in table 1 (chart) [Iheagwara et al. 2000], 100% utilization could be 64 byte frames at 14,880 pps, or 1,518 byte frames at 812 pps. There is a big difference here because processing-wise, the two are not equal. This cannot be related the above to capacity utilization, because less than 50% of the information required to simply say that utilization is "X" Mbps is available.

46 (64) 14,880 5,475,840
64 (82) 12,254 6,274,084
128 (146) 7,530 7,710,720
256 (274) 4,241 8,706,048
512 (530) 2,272 9,306,112
1,024 (1,042) 1,177 9,641,984
1,500 (1,518) 812 9,744,000

Table 1 Data Field Size Max Frames/sec Max Data Field Bits/sec.

Number of sessions
The complexity of analyzing IDS performance increases with another variable – number of sessions. This is because many ID systems have to track state and to a certain extent; the number of sessions is a huge factor. In this regard, 4,880 pps between two hosts is very different from 14,880pps between 5,000 hosts. This is demonstrated by the fact that there have been instances when ID systems starts degrading in performance at 6,500pps under 35Mbps network load with little chance of recovering based on the number of sessions observed by the IDS.

7.0 Interoperation Problems

There are very serious interoperation issues that affect the performance of Intrusion detection systems. Network ID systems work by predicting the behavior of networked machines based on the packets they exchange. The problem with this is that a network monitor that is not active cannot accurately predict whether a given machine on the network is even going to see a packet, let alone process it in the expected manner. The existence of a number of factors could make the actual meaning of a packet captured by IDS ambiguous. These can be considered as follows:

Network Inconsistencies: A network IDS is typically on an entirely different machine from the systems it's watching. Often, the Intrusion detection systems are at a completely different point on the network. The basic problem facing a network IDS is that these differences cause inconsistencies between the ID system and the machines it watches. Some of these discrepancies are the results of basic physical differences, others stem from different network driver implementations. For example, consider an IDS and an end-system located at different places on a network. The two systems will receive any given packet at different points in time. This difference in time is important; during the lag, something can happen on the end-system that might prevent it from accepting the packet. The IDS, however, has already processed the packet thinking that it will be dealt with normally at the end-system.

Protocol design problems: An IP packet with a bad UDP checksum will not be accepted by most operating systems. Some older systems might. The IDS needs to know whether every system it watches will accept such a packet, or it can end up with an inaccurate reconstruction of what happened on those machines. Some operating systems might accept a packet that is obviously bad. A poor implementation might, for example, allow an IP packet to have an incorrect checksum. If the IDS don't know this, it will discard packets that the end system accepts, again reducing the accuracy of the system.

Denial of service problems: Even if the IDS knows what operating system every machine on the network runs, it still might not be able to tell just by looking at a packet whether a given machine will accept it. A machine that runs out of memory will discard incoming packets. The IDS has no easy way to

determine whether this is the case on the end-system, and thus will assume that the end-system has accepted the packet. CPU exhaustion and network saturation at the end-system can cause the same problem.

Together, all these problems result in a situation where the IDS often simply can't determine the exact nature of a packet or implications of a packet merely by examining it; it needs to know a great deal about the networking behavior of the end-systems that it's watching, as well as the traffic conditions of their network segments. Unfortunately, the current IDS architecture is short on this and a network IDS doesn't have any simple way of informing itself about this; it obtains all its information from the packets it captures during attack detection.

8.0 Security Problems in Routing Protocols

The effectiveness of IDS operation and functional performance is closely tied to the performance of routing protocols. This is because the Intrusion detection systems rely routing protocols as a transport mechanism. Thus, designing effective IDS entails defining interoperability issues with routing protocols.

The present weaknesses in the implementation of the TCP/IP stack, a major transport mechanism in enterprise network systems, manifests in different attack forms (see table 2). These essentially are attacks using ICMP messages which includes: Denial of Service (DoS) (Figure 2) via ICMP messages, Re-routing with ICMP Route Redirect, ICMP Router Discovery messages, and ICMP informal messages such as those used in Ping of Death and Smurf attacks.

Attack families	Example	Network-based approach	Host-based approach
Denial of Service	SynFlood Attack	Best solution	Poor strategy
Scanning and Probing	ICMP	Best solution	Good solution
Password Attacks	L0phtCrack	Poor strategy	Good solution
Privilege Grabbing	Buffer Overflow		Best solution
Hostile Code Insertion	Malformed URL	Poor strategy	Best solution
Vandalism	Melissa Virus	Poor strategy	Best solution
Proprietary Data Theft	Targeting Key Sources		Good solution
Fraud, Waste, and Abuse	BO2K	Poor strategy	Good solution
Audit Trail Tampering	Covering a Trail		Best solution
Security Admin Attacks	Backdoor insert		Best solution

Table 2. Grouping of network exploits into attack families.

Attacks on routing protocols could come from both within and outside the network. Outside attacks masquerade as routers that distributes fabricated, delayed or incorrect routing information while inside attacks are mounted by a subverted or compromised router. Such attacks may have serious consequences on the network infrastructure and on end-to-end communications. Feeding false routing

information into an autonomous system (AS) may compromise the routing table of some of the AS routers. This will result in DoS on the hosts which trust that router with the implication that some hosts may not be able to reach some legitimate destinations, or the traffic flows for some particular destinations are deviated through sub-optimal routes. The packets, which follow routes that subverted, routers indicate, may be subject to eavesdropping and modification.

The manifestations of these attack sets are at the forefront of enterprise network security. This is more so because of the recognition that the responsibility for maintaining network connectivity falls on routing protocols, making it evident that routing security is an essential issue for the entire network infrastructure. Instantiations of these are attacks are given in table 3.

Attack name	Attack mode
Time, IP and UDP spoofing	Network directed
ARP and untrusted node	Network directed
ICMP sweeps (pre-attack probe)	Network directed
DNS Zone Transfer (pre-attack probe)	Network directed
Sendmail	Network directed
FTP daemon	Network directed
NNTP	Network directed
MBONE / Multicasting	Network directed
X11	Network directed
Port Scan (pre-attack probe)	Network directed or system based
Denial of Service	Network directed
Operating System Attacks (unauthorized access)	System based
Hostile code insertion	Network directed or system based
Root Kit Attacks	System
Privilege grabbing	Network directed or system based
Finger and Whois	Network directed

Table 3. Instantiations of common attack
TCP/IP attacks

Instantiation of these attacks come in different forms. For example, the Internet control message protocol (ICMP) a "best – effort" service used by IP nodes to report errors encountered while processing IP datagrams and to perform other network layer functions, such as diagnostics and monitoring have been used to launch different types of attacks such as DoS figures 2.

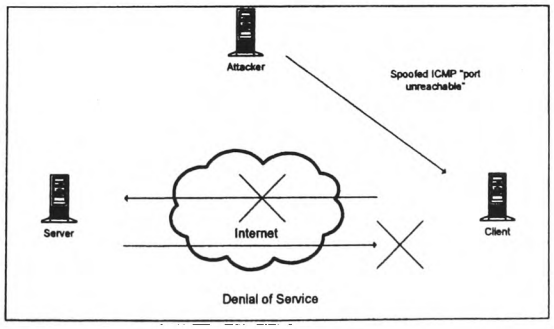


Figure 2. Topology of a DoS attack

Consequently, for an IDS design to be effective, it must incorporate into the design process all safeguards to protect its reliance on transport protocols. This is extremely complex to define, delineate or workaround because of the constantly changing attack system topology. Suffice it to say that a limited remedy must include devising a good validation program for recreating known instantiations and implementing proactive programs that are used to identify issues at TCP/IP (e.g., ports scan) and routing protocol levels and incorporating sophisticated system debugging programs that attackers use to map enterprise networks.

9.0. Incorporation of Basic Principles into IDS Design.

Justification of any security policy for the design of IDS should be aimed to address the flaws in the current implementation.

The response to potential threats involves operational requirements analysis, risk analysis, system design support evaluation, and related access control modeling and analysis, and the evaluation of secure communications services and systems. The clear and unambiguous definition of specific security needs, a crucial step in security engineering must receive due emphasis in system security policy formulation.

Contemporary research on designing secure systems focus on several key issues with the primary objective being security. System designs revolve around the security policy while providing a large amount of functionality. Another issue is integrity, where system designs attempt to ensure data integrity throughout the system. Additional issues include resource management, performance and user interface.

The IDS architecture is designed with the above in mind. However, due to the complex nature of the environments where the IDS is deployed, several pitfalls are constantly experienced bringing to focus the question of whether proven formal methods were used in the development and design process.

The design of secure systems rely to a great extent on using appropriate methodologies to prove their correctness and efficacies throughout the life cycle of the design.

It is known that following a good design practice could reduce defective designs. Bugs in a design that are not uncovered in early design stages can be costly, and bugs that remain undetected until after the design is completed and deployed can be extremely expensive. The use of formal methods, i.e., the application of mathematical methodologies to specification and validation of systems, can aid in tackling these challenges.

The planning and design of secure systems draws from known safeguards against every known and potential threat. In the case of IDS, its main intent is to conceptualize the most appropriate response to the threat given current or the projected availability of appropriate countermeasures. In order to accomplish this task within various overriding security requirements, considerations begin to focus on risk and countermeasure cost trade-offs, including the cost of long-term maintenance and reliability requirements. Often conceptualized system development is different from the real world development model. Theoretically, the assumptions made might be correct but because of difficulties in vision and policy execution there are differences.

Often the design of effective systems entails adopting a complementary approach to encapsulate experience of good and bad practice into empirical rules. For instance, in the field of cryptographic protocol design, the robustness principles are helpful, because adherence to them contributes to the simplicity of protocols and avoids a considerable number of published confusions and mistakes. Anderson and Needham [20] propose a number of robustness principles, and Abadi and Needham [6] introduce complete analyses of desirable protocol properties and relevant limitations. These could be

extended to the choice of objects for execution at the different stages of the design process for complex system design, i.e., IDS.

The experiences in the operation of IDS will be extremely beneficial in encapsulating both the good and bad practices into empirical design rules. The experiences are briefly discussed in Section 6.0.

The design process entails dealing with empirical rules and, in some cases; following one design principle will sometimes lead to violating another. In addition, even following all the rules will not guarantee a sound design. For instance in the design of protocols, many authors have considered the question of what are appropriate goals in the context of protocol analysis. Accordingly, Boyd el al. [1994] reviewed some design goals in authentication protocols and proposed a classification of them: intentional and extensional goals.

Intentional goals are generally concerned with ensuring that the protocol runs correctly as specified, while *extensional goals* are concerned with what the protocol achieves for its idealized and actual models of system development participants. It has been suggested that attacks should be measured by whether or not they violate extensional specifications even if intentional ones have been used to find the attacks in the first place. Boyd proposes a hierarchy of extensional protocol goals, which includes the major proposed goals for key establishment. He furthermore demonstrated how these extensional goals could be exploited to motivate design of entity authentication protocols.

The following useful set of principles applied to other designs could be extended to the IDS design process:

- Distributability: no central coordination takes place, which means there is no single point of failure;
- Multi-layered: an existing concept that combines different mechanisms to provide high overall security;
- Diversity: in diverse systems security vulnerabilities are likely to be less widespread. This can be achieved by either making the system unique or by diversifying the protected system;
- Disposability of any system component;
- Autonomy of each individual component; and
- Adaptability of the system to different environments.

The design process consists of steps each inter-relating to the other in a specific manner with input and output components that inform the next or previous design step in devising the essential design elements necessary to complete the terminal consideration of that particular step.

System design typically starts with a high-level specification, given in terms of block diagrams, tables, and informal text conveying the desired functionality. A combination of top-down and bottom-up design techniques is applied until a final design is obtained.

One item specific to the specification stage is the initial analysis also called risk assessment of every facet of the enterprise network at the system, host and network levels. For the
enterprise, insuring high quality security for a system and host component is a major thrust of the initial planning effort. Thorough knowledge of the system will provide the input materials for a sound IDS design

The system security engineering management (SSEM) plans produced during this phase should delineate the criteria and operational environments for specific solutions to defined security needs. These needs may span the range from providing additional network connectivity to existing systems to the provisioning of new network and/or computing environments. The plans should identify specific security evaluation, implementation, and deployment requirements as needed to complete the delivery order.

In the analysis phase, crucial design implementation questions at the network levels that should define the scope of interoperation of the IDS with decomposable network components include:

- What kinds of access controls (Internet, wide area network connections, etc.) are going to be in place?
- What authentication protocols and procedures are to be used for local area networks, wide area networks and dialup servers?
- What type of network media, for example, cables, switches, and routers, are used and what type of security do they have?
- Will security be implemented on file and print servers?
- Will encryption and cryptography, Virtual Private Networks (VPNs), e-mail systems, and remote access be used over the Internet?
- What procedures will be implemented to ensure conformity with networking standards?

The future IDS design should be such that its placement on the network should not impair its ability to function effectively. The optimum performance of the current designs has been shown [Iheagwara et al. 2000] to depend on the deployment location of the IDS on the network. In this regard, in order to isolate points of vulnerability, it is necessary to analyze the data flow through the networks. From this perspective, there are two basic scenarios:

(i) Data stored on a computer: For data stored on a local computer, the operating system is the major provider of the necessary services for the protection task. Using these services requires that they be properly configured.

(ii) Data traveling across communication points: Data traveling between locations needs to be secured in a different way, and this often involves encryption. Generally speaking, this data is in one of two forms: data in the form of network packets coming into a system, and data that is leaving the system.

Protecting incoming data encompasses both guarding the data itself and guarding the system against threats posed by the data once it has entered the network. Protection activities include a system check to ensure that the data comes from an authorized sender and that it can perform only authorized tasks.

Protecting data that is leaving a computer involves insuring that it reaches its target in exactly the same format in which it was sent, without being changed. The session and data type, as well as data content, must be unreadable by a third party—that is, privacy must be preserved.

The enterprise network usually offers several possibilities for data to leave or enter a specific computer. Computers can have individual modems with a variety of available connection scenarios. Additionally most computers are connected to a local (internal) network from which data can branch through multiple points to numerous destinations.

From the security perspective, there are two major issues involved in this exchange of information: (i) the data that is leaving a computer must reach the target without being read or changed before it reaches its destination and (ii) the packets that are reaching and entering a computer must be from an authorized user and their objective must be to pursue authorized tasks.

Typical network scenarios are:

- A corporate network that shares a private network with another company.
- A corporate network with Web servers located at an ISP, accessible either via dial-up or a permanent connection.
- A corporate network with dial-up capabilities.
- A corporate network with a permanent connection to the Internet.

Given this complex scenario and the many opportunities it offers for breaches of security, implementing IDS security should be a step-by-step process that starts with the primary local resource where the data is housed, continues through the intervening points, and concludes with the permanent connection to the "rest of the world." To support this step-by-step security implementation process, a suitable analysis and deployment architecture is needed.

Data en route cannot be directly protected by services of the operating system. However, there are different technologies (protocols) available to create a tunnel between two nodes and encrypt the information, e.g., VPN. All of them have their individual limitations and the decision regarding the appropriate technology or combination of technologies needs to be well planned. This then becomes another vital IDS design input material because there must exist an interdependency relationship of these technologies, i.e., VPN and the IDS technology.

The goal of every enterprise network is growth with a motive for profit. Technically this translates to factoring in scale up parameters in the design. This is another thrust of the analysis phase of the design process. Any scale up of the network brings with it issues that will permanently or transiently affect the system thus warranting in some cases a revision of the system topology and architecture. For example, the growth of the enterprise network could mandate a redesign of the network system resulting to the reevaluation and determination of the basic elements of growth. Such could simply mean incorporating the Gigabit Ethernet in place of 10/100 MB Ethernet technology. The issue of increasing the network bandwidth mandates consideration of security at the component and network levels. This in fact has given rise to the introduction of Gigabit IDS.

High-availability and scalable bandwidth considerations are essential design goals. High availability is a function of the application as well as the whole network between a client workstation and a service located in the network. While the mean failure time of individual components is a factor, network availability is determined mostly by the network design. This means that the application of design principles in the implementation of IDS should short circuit such issues as memory and CPU usage.

Design validation is a critical design process. The purpose of validation is to determine secure design implementation weaknesses. This involves ascertaining that the physical design does indeed meet its specification. In a traditional design flow, this is realized through simulation and testing. Because testing for nontrivial designs is generally infeasible, testing provides at best only a probabilistic assurance. Formal verification, in contrast to testing, uses rigorous mathematical reasoning to show that a design meets all or parts of its specification. This requires the existence of formal descriptions for both the specification and implementation. Such descriptions are given in notational frameworks with a formal semantics that unambiguously associates a mathematical object with each description, permitting these objects to be manipulated in a formal mathematical framework.

Issues pertinent to IDS design specification; verification and validation are discussed in the next section.

10. Specifications and Verification

The security property specification contains the information needed to validate a system and must aggregate system intrinsic and extrinsic properties. Thus, the specification of the secure communications capabilities or transport mechanism could be seen as performing robust validation and Quality Assurance exercises based on the defined requirements and evaluation criteria. Of importance is the demonstration of interoperability between components (hardware and software) of the system. The component level operational interoperability evaluated in this area should at a minimum honor known security constraints.

The security property specification of the IDS design should be defined after preliminary performance specifications for software, hardware; and network topology, architecture and subcomponents are prepared. Such properties must evolve to satisfy safeguards of identified threats known from operational experiences and vulnerabilities processed through system design modifications and risk management. Each adversarial threat is modeled and examined in terms of the capabilities of the countermeasure to be employed. Using a criterion, the alternatives are sequentially evaluated and accepted or discarded based on their current relevance to the protected information or attack scenario.

Specification and verification generally requires that some assumptions be made on the behavior of the environment in which a device is intended to operate. If actual operating environment violates these assumptions, the device may fail despite successful verification.
Specification and verification of system design can be accomplished using different techniques that specify a set of constraints to satisfy. These constraints usually specify how the system handles concurrent access dependencies.

There are two main approaches to the specification and corresponding verification. The first is concerned with specifying desired properties for the design. Formal verification is concerned with properties of temporal nature, i.e., they do not pertain to static attributes of the system but rather characteristics of the system behavior or execution such as network traffic characteristics. Temporal logics are unifying framework for expressing such properties. Verification amounts to showing that all of the system's possible behaviors satisfy the temporal properties of its specification.

The other approach is based on specification in terms of a high-level model of the system. In this case, the behaviors of a system are given by a set of all behaviors of the higher-level model, rather than a set of temporal properties. Verification then requires showing that each possible behavior of the system's implementation is consistent with some behavior of its high-level specification.

The combination of the two approaches is common: First, a high-level model of the design is shown to satisfy a set of desired temporal properties. Then a series of more and more detailed specification are developed, each of which is an implementation of the specification at the next higher level. In an appropriate technical framework, the temporal properties of the highest-level model are preserved by the refinement steps and thus are satisfied by the lowest, most detailed, level. In this context, the first type of verification is also referred to as "design or property verification", while the second form is known as "implementation verification." The two are conceptually the same since they are verification instance of the same problem: the specification defines some constraint on the allowed behaviors of a system, and verification requires showing that the implementation meets this constraint. The two are relevant to the IDS verification process.

Because of the complexities of the design process, it is helpful to verify the effectiveness of the IDS design from the formal security property specifications that govern the mechanisms and functionality of the interconnected components and the IDS interprocess communication.

For instance, a secure IDS design must include verification of a set of specifications of entropy security values which should be hard coded into component (detection modules) and system level designs such as those used to protect transport protocols against known vulnerabilities like IP spoofing, UDP Spoofing, DNS and Zone Transfer, etc.

Requirements for validating secure IDS designs should be drawn from past implementation experiences. The validation testing should incorporate techniques known to be effective in discovering design flaws. The identification of interoperation defects of the design using different probing techniques (Appendix 3) with the transport mechanism (routing protocol) is an essential requirement. In this case, the first thing to do is to identify a network segment by setting up a network analyzer and collecting some traffic on that segment. Analysis of the interaction of the detection ability of the complex traffic stream may help identify potential design problems. There are a variety of proprietary network protocols that can accomplish this.

Another requirement is taking inventory of all of the network software in order to map out probing specifications. This could be accomplished in a short time frame. This includes identification of a variety of programs and protocols for parameterizing the traffic stream that is sought after in the test.

Depending on the situation and the available information, it can be very difficult to get a clear picture of all aspects of a security event on the network. Distinct events may not seem related until another piece of the puzzle is added for clarity. Attempting to answer the basic questions about system components or events such as vendor provided component and software specifications in order to determine functionality and performance parameters is a good place to start and this should provide the framework to paint a

picture of what will transpire during the test, deviation of which is a trigger for insecure design and configuration. Thus, knowledge of the functional and performance specifications of system components is an essential requirement.

The testing process also entails using design objects, e.g., firewalls, routers and switches as targets (figure 3) for analysis of event streams. These objects as well as simulated event streams such as attack sets with parametric specifications are essential validation requirements. This will aid in the determination of threats and object threat levels once the target has been identified and the event stream set.

There is also the requirement to delineate and characterize the impact that a vulnerable system and/or its component will have on the network when in operation. This will help to determine what design remedies need to be applied. This is a precondition towards determining if the defect is a single point of failure for the network and if it could be remedied by making certain changes in the design.

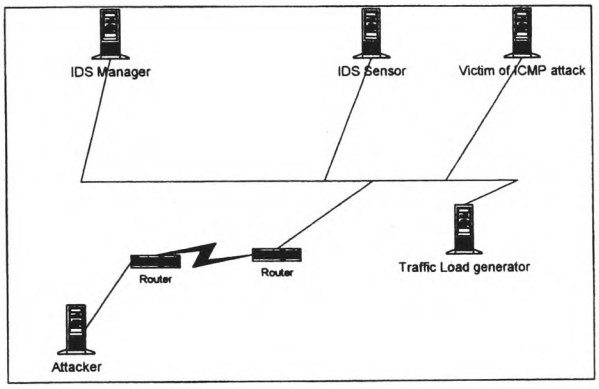


Figure 3. Typical topology for validation testing.

11. Conclusion

This paper has discussed a few examples of the many possible issues associated with the effective design of secure systems that could be related to IDS design. As was shown, security issues manifest in virtually all aspects of system design from conceptual specification through verification to operation.

We presented the basic architecture of the IDS design, explored the implementation and design pitfalls. We reviewed the application of formal methods in secure designs using the development of transport protocols as an illustration. Further, we presented a few problems associated with the implementation of transport protocols.

Drawing from practical experiences in the implementation of the IDS, we discussed and analyzed associated design issues. This includes interoperability problems resulting from complex networking and operating systems technologies and the constantly changing landscape of enterprise security policies.

A veritable design goal is accurate verification. Design models can vary and do infact range from abstract models where the design is divided into a few blocks, down to very detailed descriptions that include the minutest component. Verification at high-level abstraction does not prove that the lower-level details of the implementation are correct. At the same time, formal methods can vary and be very effective at finding errors at high levels of abstraction before a large design effort is invested in implementing a flawed system architecture's network.

The basic concepts underlying design verification were explored from the framework of proven theorem. Many examples using theorem provers verify that some model of a design is a refinement of a higher level, although deductive proofs of temporal properties are also possible.

We presented different specification and verification techniques and related them to actual system designs.

We proposed informal validation program (testing and simulation) that involves writing specifications for each class of design components using known issues in the implementation of the IDS – primarily transport protocols issues to create attacks for the modular validation of the design.

This work demonstrates that the correctness of the IDS architectures and implementation standards can guarantee its design functional effectiveness and is realizable if a methodical design approach based on formal and other methods is followed. Most notably the IDS interprocess communication, is a crucial design element because it's verification through layers of components that guarantee the correctness of system events that makes reference to operating system calls, of the operating system calls in terms of network calls, and of the network calls in terms of network transmission steps is decisive to the IDS functional effectiveness.

Concluding, it has become widely accepted that established standards and design rules must be taken into account distinctively and in combination in a complementary way during all phases of the design process in order to attain effectiveness and reliability of the security schemes for the enterprise network.

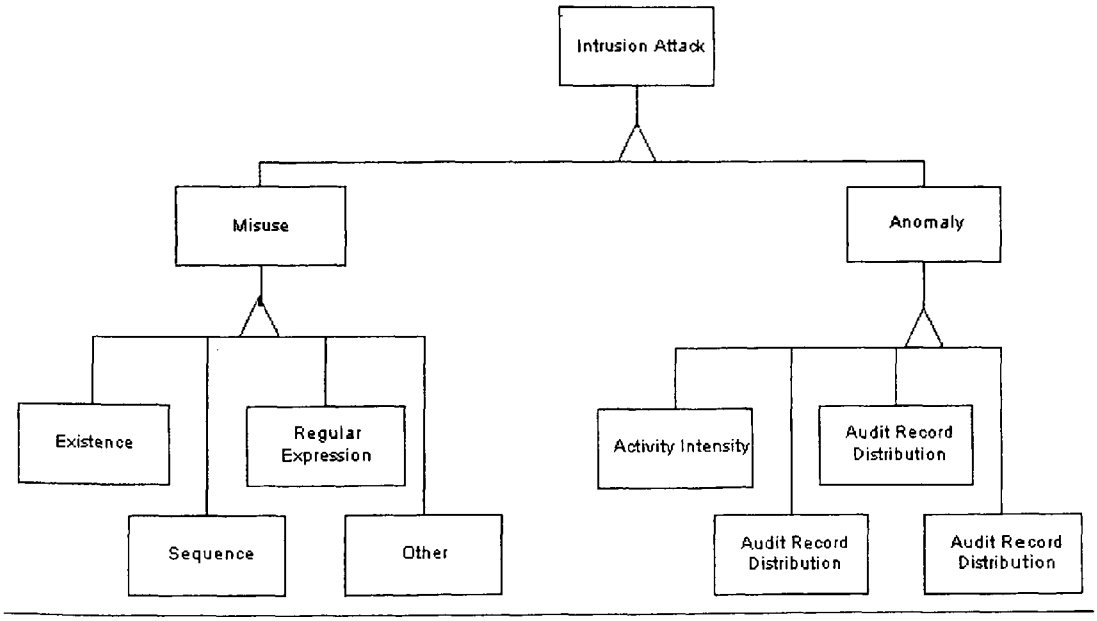
References

1. Abadi M., and Needham R. 1994. Prudent Engineering Practice for Cryptographic protocols. In *Proceedings of the 1994 IEEE Symposium on Security and Privacy*, 122-136, IEEE Computer Society Press.
2. Anderson R., and Needham R. 1995. Programming Satan's Computer, *Computer Science Today: Recent Trends and Developments, LNCS 1000*, 426-440, Springer Verlag.
3. Boyd C., Mao W. 1995. Designing Secure Key Exchange Protocols, In *ESORICS '94, Proceedings of the Third European Symposium on Research in Computer Security*, (1994) 93-105, Springer Verlag.
4. Buttyan L., Staamann S., Wilhelm U. 1998. A Simple Logic for Authentication Protocol Design. In *Proceedings of the IEEE Computer Security Foundations Workshop XI*, 153-162, IEEE Computer Society Press
5. Denning D.E., 1987. An intrusion-detection model. *IEEE Transactions on Software Engineering*, 13(2), February 1987.
6. Desmedt Y., Burmester M., Millen J. 1997. Design vs. Verification: Is Verification the Wrong Approach? In the *Proceedings of the 1997 DIMACS conference.*
7. Gollmann D. 1996. What do we mean by Entity Authentication, In *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, 46- 54, IEEE Computer Society Press.
8. Gong L., Syverson P. 1995. Fail-Stop Protocols: An Approach to Designing Secure Protocols. In *Proceedings of DCCA-5 Fifth International Working Conference on Dependable Computing for Critical Applications*, 45-55.
9. Heintze N., and Tygar J. 1995. A Model for Secure Protocols and their Compositions. In *Proceedings of the 1994 IEEE Symposium on Security and Privacy*, IEEE Computer Society Press.
10. Iheagwara, and A. J. C. Blyth, 2002. Evaluation of The Performance of ID systems in a switched and distributed environment: The RealSecure Case Study. *International Journal of Computer Networks and Telecommunications (Computer Networks)*, Vol 39 (2002) 93-112
11. Iheagwara C, and Blyth A, Singhal M. 2002. A Comparative Experimental Evaluation Study of Intrusion Detection System Performance in a Gigabit Environment, *Journal of computer security*, June, 2002

12. Jakobson G and Weissman.M.D. 1993. Alarm correlation. *IEEE Network*, pages 52-59, November 1993.
13. Lunt T.F, Jagannathan R, Lee R, Whitehurst A, and Listgarten S. 1989. Knowledge-based intrusion detection. In: *Proceedings of the AI systems in Government Conference*, March 1989.
14. Maimon U. Port scanning without the SYN flag. *Phrack Magazine*, vol. 7, Issue 49. 1985.
15. Meadows C.1995. Formal Verification of Cryptographic Protocols: A Survey, Advances in Cryptology, *Proceedings of ASIACRYPT '94*, 133-150, Springer Verlag.
16. Milner R., Parrow J., Walker D. 1992. A calculus of mobile processes, *Information and Computation*, 1-77.
17. Morris R.T. 1985. A weakness in the 4.2BSD UNIX TCP/IP software. In *Computing Science Technical Report 117*. AT&T Bell Laboratories, Murray Hills, NJ, 25 February 1985.
18. Mounji A, Le Charlier B, and Zampunieris D. 1995. Distributed audit trail analysis. In: *Proceedings of the ISOC 1995 Symposium on Network and Distributed system Security*, pages 102-112, February 1995.
19. Nessett D. 1990. A Critique of the BAN-Logic. ACM Operating Systems Review, Vol. 24 No. 2, 35-38.
20. Porras P.A and Valdes A, 1998. Live Traffic Analysis of TCP/IP Gateways, In *Internet Society's Networks and Distributed systems Symposium on Security*, March 1998.
21. Richards K, 1998. Network Based Intrusion Detection: a review of technologies, *Computers & Security*, 18 (1999) 671-682.
22. Rudolph C. 1998. A Formal Model for Systematic Design of Key Establishment Protocols, In *ACISP'98 Proceedings of the Third Australasian Conference on Information Security and Privacy*, 332-343, Springer Verlag.

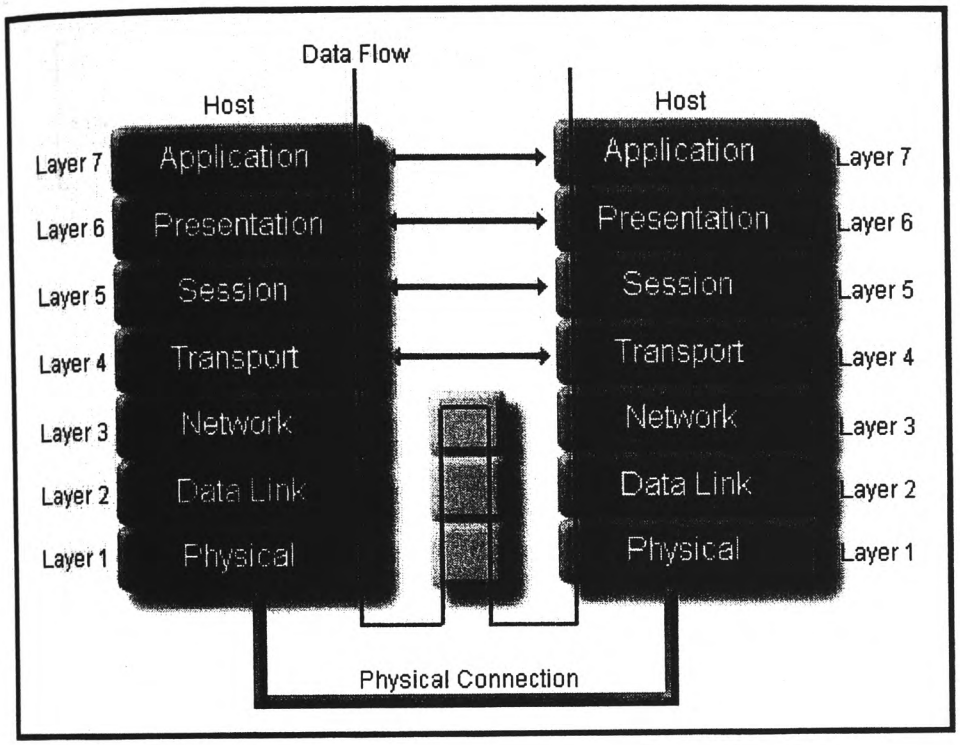
Appendix 1: The class hierarchy of Intrusion attacks.

Intrusions fall into two categories, namely Misuse and Anomalous Behavior. The Figure below shows the attack hierarchy. The forks below the rectangles represent inheritance.



Appendix 2: Open Systems Interconnection (OSI) Reference Model

The OSI seven-layer reference model is an attempt to rationalise the complexity of computer communications by standardising the functions into seven layers. Each layer is defined by the service it provides to the layer above. From an end users perspecitive, the applications they use sit above the top layer and use its facilities. The physical communication medium is underneath the bottom layer.

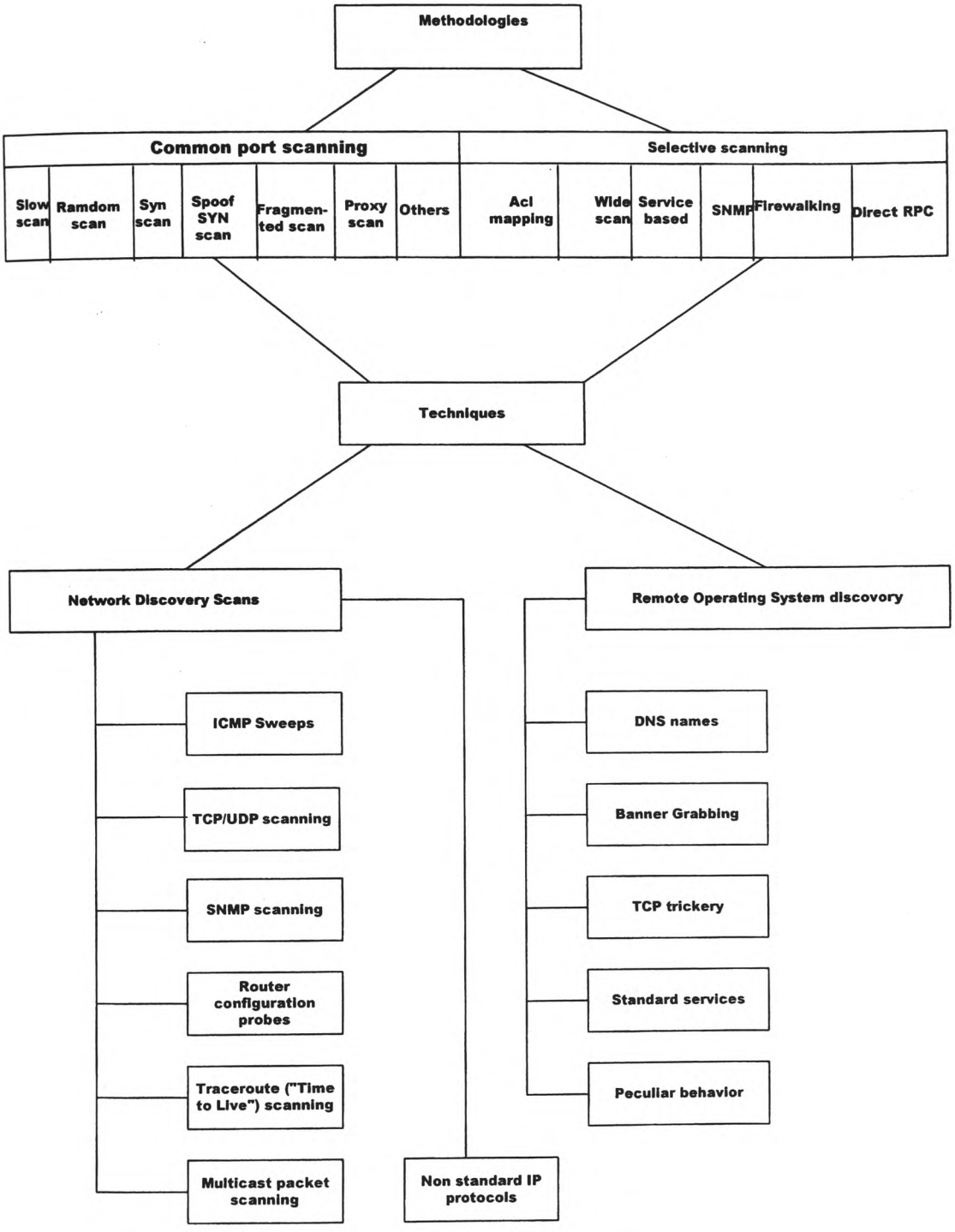


The OSI reference model

Summary of Layers

1. **Physical Layer:** Transmission of bits through the physical medium
2. **Data Link Layer:** Exchange of data between two nodes on a network
3. **Network Layer:** Routing and Connection control of the network
4. **Transport Layer:** Transparent end-to-end transmission of data
5. **Session Layer:** Logical flow of data between communication end points
6. **Presentation Layer:** Translates and formats the data
7. **Application Layer:** Provides end user services

Appendix 3: Validation test-probing scheme



Appendix 3

"The impact of security layering on end-to-end latency and system performance in switched and distributed e-business environments." Computer Networks Journal Vol 39-5

ELSEVIER

Computer Networks 39 (2002) 827–840

COMPUTER NETWORKS

www.elsevier.com/locate/comnet

The impact of security layering on end-to-end latency and system performance in switched and distributed e-business environments

Charles Iheagwara *, Andrew Blyth

8715 First Avenue, 1413D, Silver Spring, MD 20910, USA

Received 17 January 2001; received in revised form 6 December 2001; accepted 7 March 2002

Responsible Editor: M. Singhal

Abstract

Contemporary e-business networks are increasingly implementing the multi-layer security scheme in order to provide a reasonable measure of security for their information systems. The implementation entails formation of a layered architecture (concentric security layers) using packet and application-level filters neither of which provides complimentary functions. The layered architecture provides convenient abstractions and increases the end-to-end latency that results into sub-optimal system performance. In this paper, we present the results of the experiment to quantify the latency introduced by security layering on end-to-end latency and investigate the resulting degree of sub-optimality of system performance in a distributed and switched e-business network.

Keywords: Network security; Latency; Authentication; Retrieval

1. Introduction

The requirements [1] for performance, reliability, speed and operational support of e-business activities in contemporary corporate information systems are increasingly becoming complex and extremely high. In terms of reliability, the system must be designed to ensure system-level availability of 99.999% on a 24×7 basis. In terms of operational support, the system must meet all of the requirements to be certified for operation. These requirements must also incorporate security schemes into the product design as a precursor to meeting all functional requirements established for the system. The implementation of the security scheme should be able to support these requirements in a manner that does not impede vital system performance indexes such as desirable low values for end–end latency, Web request-response time, network throughput and protection of the privacy of data.

To realize the above, stringent security measures such as implementation of the multi-layer

* Corresponding author. Tel.: +1-3015871236.
E-mail address: iheagwarac@aol.com (C. Iheagwara).

1389-1286/02/$ - see front matter © 2002 Published by Elsevier Science B.V.
PII: S1389-1286(02)00250-5

security scheme is widely adopted. The scheme envisages the use of a combination of packet filters and application-level firewalls because neither the packet filter nor the application-level filters provide complimentary functions. The implementation requires the formation of security layers using packet-forwarding devices with varying degrees of packet filtering and blocking functions. A typical arrangement is the use of filtering routers at the perimeter of the network and application-level firewalls inside the network. Also, part of the stringent security measure is the deployment of intrusion detection systems (IDSs) to detect unwanted traffic.

Intrusion detection as an important component of a security system, complements other security technologies. By providing information to site administration, an IDS allows not only for the detection of attacks explicitly addressed by other security components (such as firewalls and service wrappers), but also attempts to provide notification of new attacks unforeseen by other components. IDSs also provide forensic information that potentially allows organizations to discover the origins of an attack. In this manner, an IDSs attempts to make attackers more accountable for their actions, and, to some extent, act as a deterrent to future attacks.

The IDSs detection modules are deployed at strategic locations across the enterprise network in order to stop attacks, misuse, and security policy violations before damage is done. When an IDSs detects unauthorized activity, it can respond in a number of ways, automatically recording the date, time, source, and target of the event, recording the content of the attack, notifying the network administrator, reconfiguring a firewall or router, suspending a user account, or terminating the attack.

Packet filtering routers are generally the smallest and the simplest form of firewall [2]. They can provide a low-cost and useful level of firewall security. Their sole purpose is to check the source address, destination address and ports in individual IP packets. Packet-filtering firewalls work by dropping packets based on their source or destination addresses or ports. They make decisions only from the contents of the current packet. Filtering can be done at input time, at output time or both depending on the type of the router. Because they only perform cursory checks on the source address, there is no real demand on the router. It takes little time to identify a bad or restricted address. The administrator makes a list of the acceptable machines and services and a stop list of unacceptable machines or services.

Packet filtering routers alone are inadequate to implement stringent security requirements. The reasons for these are:

1. A packet filter does not enforce transport-level issues, such as the early 1997 attacks against Windows NT (invalid TCP window size, invalid sequence number). One relies on the software running on internal systems for security, and as previously mentioned this is typically the weakest point. For example, any type of packet filter will pass through an SMTP connection to an internal mail server. None are able to filter out known problem areas before sending it onto an internal Mail Hub. An application gateway, such as the Gauntlet firewall, can and does.
2. A stateful packet filter keeps state information about connections. It may let the entire packet go out just as it came in (or visa versa), so long as it matches the rules. A stateful packet filter typically does not examine the data. It does not talk on behalf of anything/anyone as a proxy does. When using an application gateway, you do not need to imitate TCP/UDP/ICMP handling because real handling is done by the firewall.
3. Current stateful packet filter implementations do not rewrite packets. The internal network is exposed to packet-based attacks. Packets are forwarded based on security rules. No packets are forwarded by application gateways. New connections are established.
4. Packet filters log much less information than application gateways. A packet filter will log source and destination addresses, accepts, and rejects. For example, HTTP connections will show single packets not filenames, URLs, number of packets, etc.
5. Packet filters are less granular (look less deeply into the communication stream) and do less security work than application gateways. Therefore, they are insufficient for applications that require a much tighter security.

On the other hand, application-level firewalls gives the network manager complete control over each service, as well as over which services are permitted. Using proxies, application-level firewalls provide much finer control over which packets can be transmitted across the firewall. Application-level firewalls can support string user authentication, provide detailed logging information, and the filtering rules for an application-level firewall are much easier to configure and test than those for a packet-filtering router. The other advantages of application-level firewalls are that they:

- do not allow any direct connections between internal and external hosts, i.e. lack of IP forwarding;
- support user-level authentication;
- analyze application commands inside the payload portion of data packets (whereas the stateful packet filters systems do not); and
- are able to keep comprehensive logs of traffic and specific activities.

The disadvantages are that they:

- are slower than packet filters;
- require the internal client to know about them;
- do not support every possible type of connection, i.e. that a proxy application must be created for each networked service. Thus, one is used for FTP, another for Telnet, another for HTTP, and so forth; and
- the last disadvantage is a factor of the level of security desired by the organization using the firewall.

To address the limitations of both application-level firewall and filtering router, the multi-layer security scheme that incorporates both types is increasingly becoming a popular implementation. However, the scheme is associated with the addition of latency from the authentication process that increases the value of the end-to-end latency. By how much the latency (network transition time) is increased and the associated performance overhead is the subject of this research.

In the main experiment, we measure the latency introduced by security layering and quantity the contribution to the end-to-end latency. Further, we characterize the effects on system performance and as well measure the degree of sub-optimality in system performance.

The rest of this paper is organized as follows. In Section 2, we discuss the related works. Section 3 presents the experimental details. The results are interpreted and analyzed in Section 4. In Section 5, we present the conclusions of this study and the recommended future work.

2. Related work

Improvements in critical system performance indexes such as network transition time, latency and network throughput have been at the forefront of numerous research works. Researchers have been fielding different methods to improve worldwide web latency. It has been recognized that there are generally two sources of worldwide web latency:

1. Network delay. Retrieving a document from a Web server using the HTTP 1.0 involves at least two round trips between the client and the server.
2. Request processing time. For each file requested, a web server has to read it from its disks into a buffer to the client.

This entails that latency can be reduced either by reducing network delays or by reducing server query/request response/processing time. HTTP enhancements and the use of different caching techniques on proxy servers and clients have been the focus of previous research works on the reduction of network delay. Improvement in Web servers' throughput has been made possible by incorporating cooperative servers, i.e. using multiple Web servers and server-side caching which is caching documents in a Web server's address space to increase Web response rate. Three caching techniques namely server-side caching, client-side caching and proxy caching have emerged as benchmark solutions.

Server-side caching strategies focuses on the reduction of servers' response time and improvement

in throughput. Arlitt and Williamson [3] analyzed access logs from different Web servers and identified ten variances among workloads. Kwan et al. [4,5] described NCSA's Web caching research that uses AFS to cache documents in Web servers' local disks. Markatos [6] proposed the notion of memory caching of Web documents, the benefit being that it reduces the number of disk accesses. By keeping most frequently accessed files in main memory, many of the requests can be served without touching the file system thereby reducing access latency.

Cunha et al. [7] who researched client-side caching analyzed access patterns of individual users and found two access patterns. First, WWW clients tend to access small files. Secondly, keeping small files in clients cache is better than keeping small ones since the former has a high latency-savings-per-byte rate. Bestavros et al. [8] compared three caching levels at the client side: the session level, the host level, and the LAN level. According to their experiments, the LAN-level caching is the most cost-effective. The reason being that for a specific document, a LAN cache keeps only one copy of it while the host and session levels keep one copy of the document in their respective caches and therefore waste more cache space.

The limitations and potentials of proxy caching have been shown by Abrams et al. [9] in their evaluation studies to be the upper bound of the hit ratio of a proxy cache that is in the range of 30–50%. Since the basic idea of caching is to move the data that the clients need closer to them. Some proxy-cache researches take geographic distribution of clients request into account. Williams and coworkers [10] compared the performance of different caching policies and implementation and found that the widely used WWW caching policy, LRU, results in poor performance. Additionally, they provided insights into other proxy cache implementations: the CERN cache [11], the Lagoon cache [12], the Harvest cache [13], and the Squid cache [14].

Martin and Russell [15], Martin et al. [16], Simpson and Alonso [17] and Tomasic and Garcia-Molina [18] also studied caching in distributed architecture. The client caches data so that operations data are not repeatedly sent to remote servers. Instead, the client locally performs frequent operations. The use of caching is most beneficial for systems that are distributed over slow networks or that evaluate queries slowly.

Gruber et al. [19] analyzed the challenges of realizing a prefix-caching service in the context of IETF's real-time streaming protocol, a multimedia streaming protocol that derives from HTTP. Their study explored how to avoid several round-trip delays by caching protocol information at the proxy server. In addition, they discussed how caching the partial content of multimedia streams introduces new challenges in cache coherency and feedback control.

Mogul [20,21] conducted research in HTTP improvement and proposed two mechanisms to improve HTTP latency: long-lived connections and request pipelining. The problem is that several features of HTTP interact badly with TCP two of which are that HTTP establishes a connection for each request and HTTP transfers only one object per request. In another instance, Padmanabhan and Mogul [22] use prefetching to hide the latency instead of reducing it. The argument is that the idle period of time between two adjacent requests from the same user can be used to prefetch the next document the user wants to read.

Burkowski [23] reports on a simulation study, which measures retrieval performance of a distributed information retrieval system. The experiments focused on two strategies for distributing fixed workload across a small number of servers.

Couvreur et al. [24] analyzed the performance and cost factors of searching large text collections on parallel systems. They used simulation models to investigate three different hardware architectures and search algorithms including a mainframe system using an inverted-list IR system, a collection of RISC processors using a superimposed IR system, special-purpose machine architecture that uses a direct search. Hawking [25] designed and implemented a parallel information retrieval information system called PADRE97, on a collection of workstations. The basic architecture of PADRE97 contains a central process that checks for user commands and broadcasts them to information retrieval engines on each work station.

Benchmarking Web servers is an active research area. Several benchmarks for Web servers have been developed, including WebStone [26] and SPECWeb. There are also studies on the overload behavior of the benchmarks and improvement of the benchmarks [27]. Almeida and Cao [28] used the Wisconsin proxy benchmark to compare the performance of four proxy servers. The study explored the effects of multiple disk use, low-bandwidth modem client connections and throughput on the performance of proxy servers. The study found that the latency advantage of caching proxies vanishes in front of modem connections.

The above-cited studies demonstrate that by improving the server's throughput and implementing different caching schemes, latency and Web query-response time can be reduced. However, the studies did not quantify or characterize the attributes of end-to-end latency in relation to process authentication in switched and distributed architectures with multi-layered security schemes. There is also no study on the impact of security layering on system performance that has been reported in scientific literature hence this is the primary motivation for this study. This study is unique in the sense that the experiments are conducted live on a distributed and switched e-business network in order to establish empirical values.

In the next section we describe the experiment on the empirical quantification of latency due security layering and characterize the impact on system performance.

3. Experimental work

3.1. Objective

The following are the objectives of this study:

1. empirical quantification of the latency introduced by security layering;
2. determination of the impact of security layering on end-to-end latency; and
3. determination of the resulting degree of suboptimality on system performance.

3.2. Experimental setup

3.2.1. Test beds

The test beds for the performance tests are shown in Figs. A.1–A.5 in the appendix section.

For the latency measurements, the test environment shown in Fig. A.6 (see appendix) consist of a Netcom Systems' Smart Bits 2000 chassis running OS 1.4.15 with firmware Version 2.1.24, 12 ML 7710 cards and two ML 7711 cards. All latency tests were performed using Netcom's Smart Flow Version 1.12.1 software suite. The error test was conducted with Netcom's Smart Window.

3.2.2. Web servers configuration

The Web servers are Intel-based Pentium 500 MHz systems with dual processors and 256 MB RAM. They are configured with Windows NT 4.0 (Service Pack 5) operating system and form a cluster farm with load balancing. The super fast caching scheme is implemented on the Web servers.

3.2.3. Client configuration

The client test server is a Dell brand Pentium 500 MHz system with dual processors, 256 MB RAM and Windows NT 4.0 (Service Pack 5) operating system. It was configured with the performance measuring software WEBSTONE and Internet connection through which the test server accesses Web site applications was established.

3.2.4. Websites description

The four Web sites are the corporate sites of a major stock exchange market hosting multimedia applications that include video graphics, database applications, and financial news contents. Each site can handle up to 10,000 concurrent connections.

3.3. Experimental method

The experimental methodology envisaged empirical quantification of the latency due to process authentication and measurement of its effects on system performance, i.e. Web query-response time. The results are then interpreted and analyzed

within the framework of environmental conditions in the distributed and switched architecture.

The experiment is conducted in two phases (parts): I and II.

- Part I of the experiment is the determination of system performance using the Web query-response speed as benchmark under the different security schemes.
- Part II of the experiment is the quantification and attributive characterization of the end–end latency and its contribution to the sub-optimal performance determined in part I. There are two sub-parts here:

  (a) measurement of latency due to switching; and
  (b) computation of latency due to authentication.

3.4. Part I: determination of web request-response speed

3.4.1. Test procedure

The client test server was configured with the "WEBSTONE" Web performance benchmark that measures the performance of a Web server.

In the test, different workloads were created on each Web server by distributing and simulating up to 85 client computers on the client test server. This simulation enabled the client test server to generate multiple files retrieval requests from the Web servers. The benchmark tests are automatically generated by the Webmaster, which used the performance measurements from the clients to generate the summary report.

Peak CPU utilization during the tests ranged from 43% to 57% for the Web servers and 49% to 53% for the client test servers. The data from the log entries was recorded over a two-week period and the test results are presented in Tables 1–3 and Fig. A.5.

Table 1
File access frequency distribution

Website number	Work-load	File size, KB	Access frequency	Access percentage
1	110 files	3135	12,730	53.645
		5045	7250	30.553
		7521	2500	10.535
		13,598	1250	5.267
2	465 files	3135	12,730	53.645
		5045	7250	30.553
		7521	2500	10.535
		13,598	1250	5.267
3	1038	3135	12,730	53.645
		5045	7250	30.553
		7521	2500	10.535
		13,598	1250	5.267
4	1578	3135	12,730	53.645
		5045	7250	30.553
		7521	2500	10.535
		13,598	1250	5.267

Table 2
File Attachments

Percentage	File attachment for all Web sites
60	No attachment
30	Doc file
0	Executable file
10	Compressed file

3.5. Part II: determination of end-to-end latency

3.5.1. Part IIa: latency due to switching

The latency of the switches was measured using Netcom Systems Smart Bits that is capable of measuring latency to an accuracy of 100 ns (0.1 ms). Multicast packets were used in the Ethernet-to-Ethernet switching. Measurements were at either layer 2 or 3 for 64-byte and 512-byte sized packets.

The throughput test determines the highest rate at which a switch can receive, process and forward packets without loss. This value is important because a pause of up to a few seconds may occur when a packet is lost from a data stream: The application—realizing data was lost—must retransmit the missing data.

3.5.1.1. Test procedure. In the latency test, two streams of FTP data, which runs over TCP, and UDP, were sent to a single switch port. The standard 100 pps test stream is used to measure latency while multiple streams of background load are applied to a set of independent ports to determine if the switch latency is a function of the background load on the switch. To assure reliability of the testing conditions, each test was repeated three

Table 3
Web query response time

Parameter	Web request processing time			
	Architecture no. 1 (s)	Architecture no. 2 (s)	Architecture no. 3 (s)	Architecture no. 4 (s)
Proxy server with caching	53 (without proxy server)	53.6	57.9 (without proxy server)	66.3
Proxy without caching	56.7 (without proxy server)	57.2	61.6 (without proxy server)	70.0
Web server farm with load balancing	53.3	53.9	58.2	66.6
Web server without load balancing	57.7	57.3	61.6	70.0

times on the switching device under test (DUT), with a cold boot of the DUT between test runs. The results of the three tests were averaged within each test situation. The latency measurements were consistent for all packet sizes and for all loads ranging from 50% to 95% at layers 2 and 3.

The throughput test is performed by having the tester send a 30-s burst of traffic through the device at half the rate theoretically possible for the given test conditions. The number of sent packets is then compared to the number received. If all were received, the data rate is increased and the trial is rerun. If all were not received, the rate is decreased and the trial is rerun. This process repeats until a rate is found at which all offered packets are forwarded.

The latency test result is presented in Table 4, Figs. 1 and 2 while the results for the throughput tests are presented (Figs. 3 and 4).

3.5.2. *Part IIb: latency due to authentication processes*

3.5.2.1. Computational method. Latency due to authentication mechanisms (L_a) was computed using the following expressions:

$$L_a = L_n - L_s$$

Table 4
Latency due to switching

Throughput, layer 2, MB	Throughput, layer 3, MB	Latency, ms
310	301.63	285
500	486.5	305
850	827.05	340
910	885.43	374

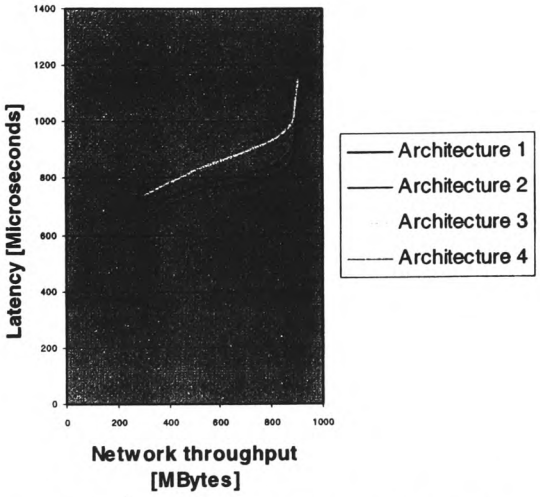


Fig. 1. Latency under the different security scheme.

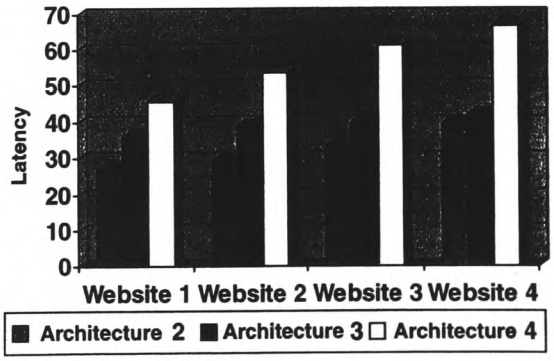


Fig. 2. Latency for the different authentication mechanisms.

where L_n is the network latency and L_s is latency due to switching.

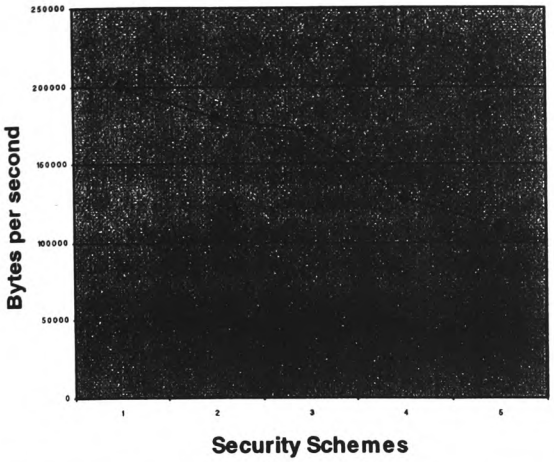


Fig. 3. Network throughput under the different security schemes.

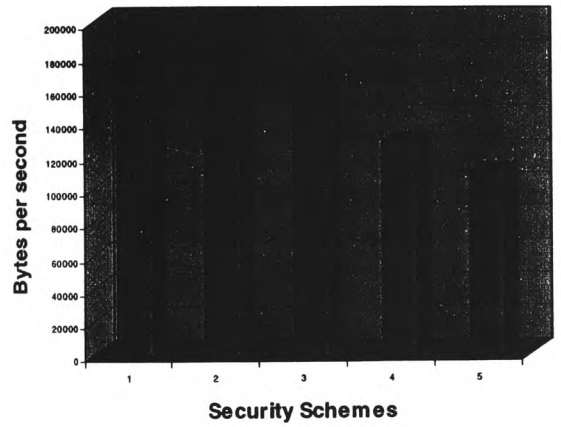


Fig. 4. Aggregated throughput capacity for each security scheme.

By substituting the different values for L_s and L_n using the above expression the latency for each given security scheme can be computed.

Latency as a percentage of overall network delay (% L_a) for each security scheme is calculated with the following expression:

$$\% L_a = \frac{L_a \times 100}{L_n}.$$

The result of the computation is shown in Table 5 and Fig. 5.

Table 5
Latency due to process authentication

Security scheme	Web site #1	Web site #2	Web site #3	Web site #4
2	27.2	29.3	33.4	39.6
3	36.1	39	40.1	42.1
4	45.9	53.9	61.2	66.9

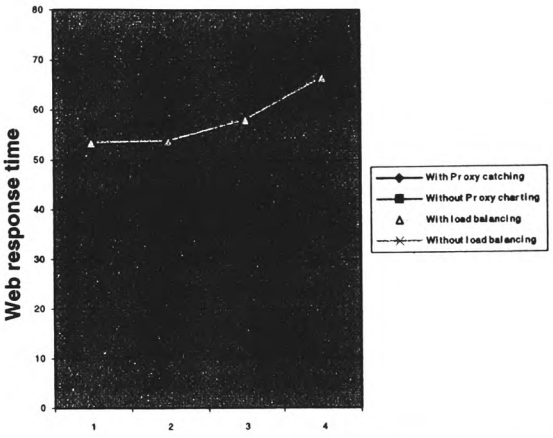


Fig. 5. Web response time under varying network conditions.

4. Analysis

Three parameters characterize the network architecture—latency, occupancy, and bandwidth. All have complicated aspects to them. Bandwidth determines how fast data can be transferred through the network interface, i.e. between the different segments and the network. For messages that carry data, node-to-network bandwidth can become the bottleneck. Occupancy has little effect on the system when the caching scheme is optimized, i.e. with maximization of memory cache and; latency significantly affects the system performance.

Network latency is the delay from the transmission of the packet header at the source to the reception of the end of packet at the destination. The latency of a message through the network depends, among other things; on how many hops the message travels in the network topology. The latency of the message from the processor to and from the network is the dominant constituent of the latency ascribable to the network topology. In

our study, the topology-related latency was assumed to be negligible because the network transit time from one node to another is always the same.

The impact of authentication latency on system performance depends not only on the structure of the authentication mechanism but also on its absolute value that is computed as a ratio of the end-to-end latency. The impact also depends on whether authentication really is the important performance bottleneck to begin with, or whether the bottleneck is some other thing, i.e. load imbalance.

For a given number of workloads, larger data sets usually improve the load balance, and hence allow the system to deliver better performance. Thus, if an application with a given workload delivers good performance on a given architecture but not on one that uses speed-limiting processes on the network, this does not in itself mean that the architecture should be replaced but rather enhanced. There may be applications, for which the less fortified architecture performs well too, and perhaps another application for which it performs almost as well as the other architecture. The question is how large are these applications relative to the base architecture that will make a major impact.

There is also the issue of the "desirable" level of performance. Typically, the larger the efficiency the larger the application size needed for a given combination of latency. Thus, the efficiency level is an important determinant of the constant factors in the expression for the required application. Furthermore, it can also affect the performance level of the required application with latency, if changing the desirable efficiency level changes the relative importance of different performance bottlenecks. For example, for an efficiency level of 30%, the dominant bottleneck to overcome by increasing the problem size may be authentication, but for a 95% efficiency level it may be load imbalance. The bottlenecks also may not behave in predictable ways as the problem size or efficiency level could change, particularly for irregular applications. In most cases, however, if the dominant bottleneck does not change, then the chosen level of efficiency will not affect the performance level required.

The impact of switching type on latency is also fundamental in defining the overall performance of Web query-response speed. High performance switches perform the role of prioritization for any application in the application-defining library (ADL) or application added to ADL as a custom entity. This means that for those mission-critical applications, we can guarantee an allocated amount of available bandwidth even during periods of network congestion. For Web-based applications, high performance switching device can distinguish between the various Web (HTTP) applications that use TCP port 80 (e.g., mainframe access via the Web, Web-based access to ERP applications, e-commerce or Web browsing) by examining the URL field. This feature facilitates readily identified e-commerce transaction flows that are given higher priority than ordinary Web browsing.

The results of our study show that there is a difference in the performance of access response time for the various security layering schemes. Throughput was high in most of the tests. In comparison with the others, the multi-layered scheme performed poorly.

The effect of authentication type on network speed (throughput) was very pronounced. In general, network throughput decreased with increase in the number of authentication negotiated. The decline as shown in Fig. 3 is largest for the multi-layer security scheme. There are five schemes represented in Fig. 3 with schemes 2 through 5 corresponding to architectures 1 through 4. Scheme 1 is the baseline network throughput.

As is evident in Fig. 5, the Web response speed under the load balancing configuration is faster than the speed under the proxy arraying configuration by as much as 6.7%.

The network throughput for the baseline configuration (network without any form of security implemented) represented as schemes #1 in Fig. 4 was 18.4 Mbps. Throughput for the multi-layered architecture was the lowest at 9.81 Mbps. The significance here is that network throughput is slowed down considerably by almost 50% in the configuration with the multi-layering scheme.

The increase in latency values for the different security schemes (Fig. 1) is more pronounced as

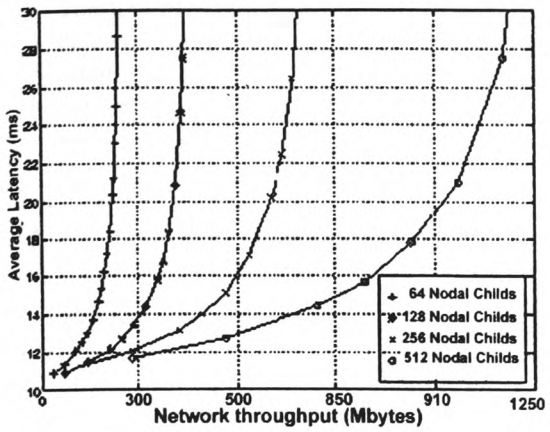


Fig. 6. Latency versus throughput for switched networks under random traffic with 64 byte packets.

the workload approaches saturation point. The increase is about 10% above the normal level. Thus, beyond saturation point, there is an accelerated increase in the latency values. Fig. 6 shows the latency of four different size switched networks under random traffic as a function of the aggregate network throughput. It clearly conforms to the findings of other researchers that there is a high increase in latency for a network load above saturation.

The latency value for the first security scheme (i.e. with only filtering router) was very negligible and as such was not represented in Fig. 2.

The interpretation and analysis of the study given above established that the throughput and authentication effects are profound and are the dominant factors for end-to-end latency.

As network approaches saturation, the latency increases by roughly a factor of 2 for the architectures with firewall and multi-layer security schemes (3 and 4). Further, latency values for the schemes (1 and 2) with filtering router and proxy servers were the lowest regardless of network load condition. This reflects the fact that communication time between the client and proxy is the dominant factor in the overall performance and it is much more important than the delay introduced by switching.

The implication is that authentication with proxy server is less time consuming than authentication with the firewall. Despite the shorter service time, network latency is lower for the experiments with lower throughput hence, it is clear that network saturation is a major bottleneck in the transmission time.

5. Conclusions and future works

5.1. Conclusions

We have described the implementation of security layering schemes in switched and distributed e-business network environments. We have conducted an experiment to quantify and characterize the end-to-end latency, and have analyzed their impact on the system performance under different security layering schemes.

Within the limits of our experiments, the following are our main findings:

- Under random traffic, latency increases linearly with increase in the workload. The degradation in the performance of Web traffic as the network size increases agrees with analytical models presented in [29]. The study predicts the throughput of switched networks under sustained random load degrade by approximately 25% from linear when the network size is increased from 64 to 512 nodes. The empirical value derived in our study is roughly 20% under similar network conditions.
- Web response time is fastest for processes without layering, slower in the single layering schemes and slowest under multi-layer security scheme.
- Latency was highest for the multi-layer security scheme where client latency increased by 50%. There is also a substantial decrease in throughput under the multi-layer scheme.
- In terms of network transmission time, the first and second layering schemes maintained roughly constant values but there is a significant decrease under the third and fourth schemes for the different network load conditions. This implies that authentication by the Cisco firewall was more time consuming than that of Microsoft's proxy server.

- The absolute value of the end-to-end latency fluctuates based on the network load. Under multi-layering, when a firewall or proxy must handle requests sent through very low bandwidth (under network throughput saturation), the time spent in the network (transmission time) dominates.
- The multi-layer architecture suffers from not being able to respond well under heavy network load. However, improvements are realized when load balancing and proxy caching are maximized.

5.2. Future works

The work presented in this paper quantified the overhead and performance problems introduced by security layering in switched and distributed networks. In future, we will investigate the techniques that could possibly combine multiple logical security layers into a single implementation to alleviate the problems associated with the current implementation that uses multiple entities to form multiple security layers.

Acknowledgements

The authors gratefully acknowledge the support of John Bass of Network world test alliance for the latency measurement tests and Frank Mercedes of UTV Computer Corporation for the performance tests.

Appendix A

The corporate network (Fig. A.1)

Appendix B

Architecture #1: under this scheme, the topology shown in Fig. A.2 incorporates a packet filtering router at the perimeter of the network.

Appendix C

Architecture #2: the topology shown in Fig. A.3 incorporates a Proxy server.

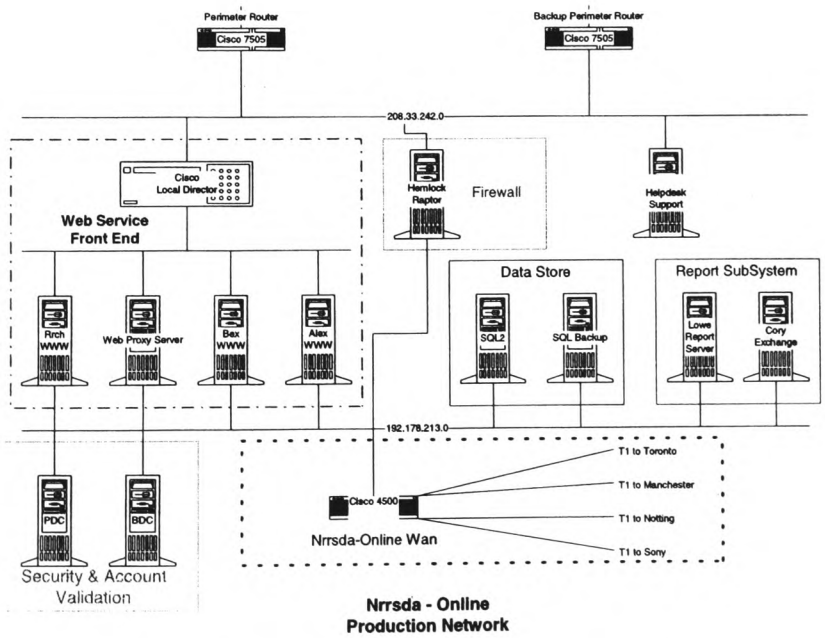


Fig. A.1. Network topology with multi-layer security scheme.

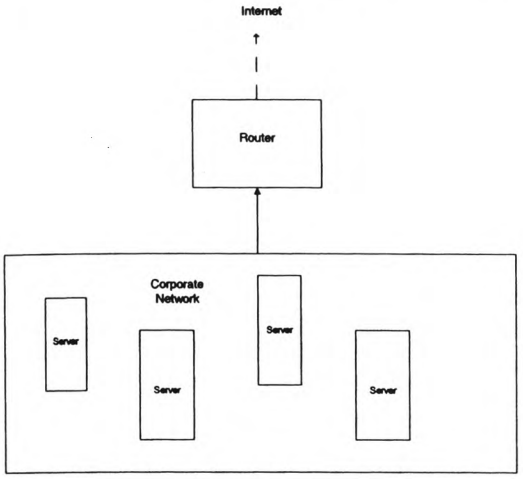


Fig. A.2. Environmental logical diagram for architecture #1.

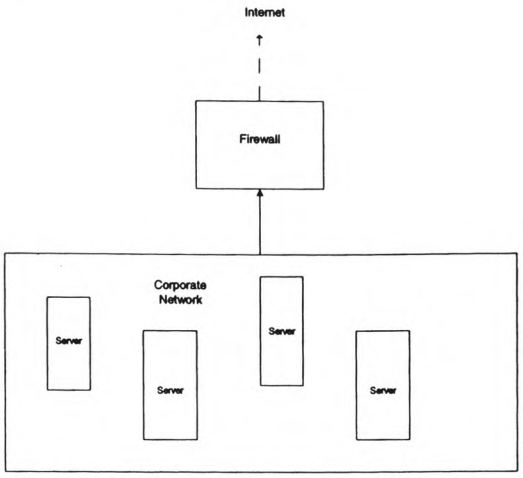


Fig. A.4. Environmental logical diagram for architecture #3.

Appendix D

Architecture #3: the topology shown in Fig. A.4 incorporates application-level filter (firewall) for process authentication.

Appendix E

Architecture #4: the setup incorporates a multi-layered scheme shown in Fig. A.5.

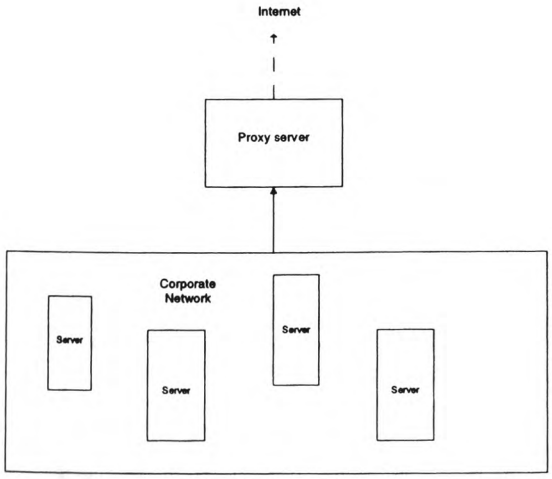


Fig. A.3. Environmental logical diagram for architecture #2.

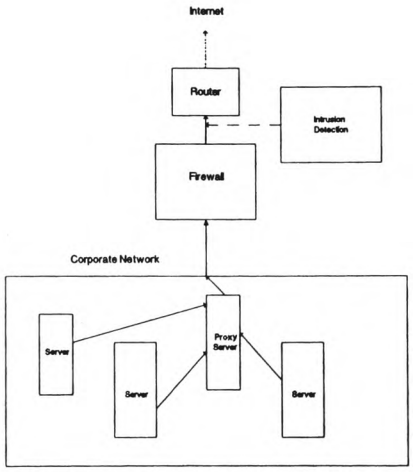


Fig. A.5. Environmental logical diagram for architecture #4.

Appendix F

Configuration for testing latency with background load (Fig. A.6).

Appendix G

Features of the network security equipments used in the tests.

Device	Vendor	Usage	Connectivity
Pix firewall 520	Cisco	Authentication of network access for applications	Ethernet Interface 100 Mbps
Router	Cisco 7505	Authentication based on access list (filtration)	Internal: 100 Mbps Ethernet Interface to Pix firewall External: T1 connection to the Internet
Proxy Server	Microsoft	Authentication based on applications	Ethernet Interface 100 Mbps for internal and external connections
Switch	Cisco 4500	Routing and Switching	Ethernet Interface 100 Mbps

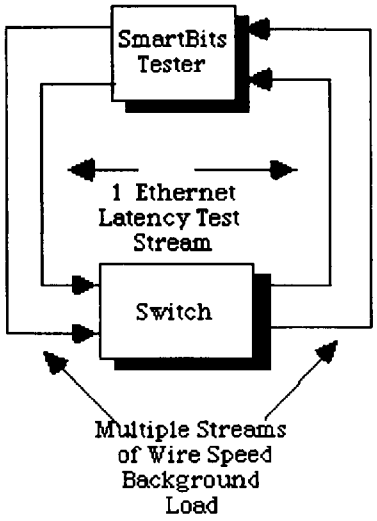


Fig. A.6. Environmental logical diagram for latency test.

References

[1] H. Harding, P. Zhong, C. Iheagwara, Optimizing the performance of enterprise networks in e-business environments, Technical document, UTV Computer Corporation, Silver Spring, Maryland, USA, 1999.

[2] M. Ranum, Firewall Performance Measurement Techniques: A Scientific Approach, Maximum Security: A Hackers Guide to Protecting Your Internet Site and Network, Macmillan Computer Publishing, USA.

[3] M.F. Arlitt, C.L. Williamson, Web server workload characterization: the search for invariants, in: Proceedings of the SIGMETRICS, Philadelphia, PA, April 1996.

[4] T.T. Kwan, R.E. McGrath, D.A. Reed, NCSA's World Wide Web server: design and performance, IEEE Computer 28 (11) (1995).

[5] T.T. Kwan, R.E. McGrath, D.A. Reed, User access patterns to NCSA's World Wide Web Server, Technical Report UIUCDCS-R-95-1934, Department of Computer Science, University of Illinois, February 1995.

[6] E.P. Markatos, Main memory caching of Web documents, in: Proceedings of the Fifth International World Wide Web Conference, May 1996.

[7] C.R. Cunha, A. Bestavros, M.E. Crovella, Characteristics of WWW client-based traces, Technical Report BU-CS-95-010, Computer Science Department, Boston University, July 1995.

[8] A. Bestavros, R.L. Carter, M.E. Crovella, Cc.r. Cunha, A. Heddaya, S.A. Mirdad, Application-level document caching in the Internet, in: Proceedings of the Second International Workshop on Services in Distributed and Networked Environments (SDNE'95), Whistler, Canada, June 1995.

[9] M. Abrams, C.R. Standridge, G. Abdulla, S. Williams, E.A. Fox, Caching proxies: limitations and potentials, in: Proceedings of the Fourth International World Wide Web Conference, Boston, MA, December 1995.

[10] M. Abrams, C.R. Standridge, G. Abdulla, S. Williams, E.A. Fox, Removal policies in network catches for World Wide Web documents, in: Proceedings of the ACM SIGCOMMM'96 Conference, Stanford University, August 1996.

[11] The CERN Proxy Cache, available at http://www.w3.org/pub/WWW/Daemon/User/Config/Caching.html.

[12] P.M.E. De Bra, R.D.J. Post, Information retrieval in the World-Wide-Web: making client-based searching feasible, in: Proceedings of the First World Wide Web Conference, Geneva, Switzerland, May 1994.

[13] C.M. Bowman, P.B. Danzig, D.R. Hardy, U. Manber, M.F. Schwartz, The Harvest information discovery and access system, in: Proceeding of the Second World Wide Web Conference, Chicago, October 1994.

[14] Squid Internet object cache, available at http://www.squid-cache.org/.

[15] T.P. Martin, J.I. Russell, Data caching strategies for a distributed full text retrieval systems, Information Systems 16 (1) (1991) 1–11.

[16] T.P. Martin, I.A. Macleod, J.I. Russell, K. Leese, B. Foster, A case study of caching strategies for a distributed full text retrieval system, Information Process Manager 26 (2) (1990) 227–247.
[17] P. Simpson, R. Alonso, Data caching in information retrieval systems, in: Proceedings of the 10th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New Orleans, LA, June 1987.
[18] A. Tomasic, H. Garcia-Molina, Caching and database scaling in distributed shared-nothing information retrieval systems, Tech. Republic, STAN-CS-92-1456, Stanford University, Stanford, CA, 1992.
[19] S. Gruber, J. Rexford, A. Basso, Protocol considerations for a prefi-caching proxy for multimedia streams, Technical document, AT&T Labs—Research, USA, September 1999.
[20] J.C. Mogul, Improving HTTP latency, in: Electronic Proceedings of the Second World Wide Web Conference: Mosaic and the Web, Chicago, IL, October 1994.
[21] J.C. Mogul, The case for persistent-connection HTTP, in: Proceedings of the ACM SIGCOMM'95 Conference on Communications, Architectures, and Protocols, Boston, August 1995.
[22] V.N. Padmanabhan, J.C. Mogul, using predictive prefetching to improve Worldwide Web latency, Computer Communication Review 26 (3) (1996).
[23] F.J. Burkowski, Retrieval performance of a distributed text database utilizing a parallel process document server, in: Proceedings of the International Symposium on Databases in Parallel and Distributed Systems, Dublin, Ireland, July 1990.
[24] T.R Couvreur, R.N. Benzel, S.F. Miller, D.N. Zeitler, D.L. Lee, M. Singhal, N. Shivaratri, W.Y.P. Wong, An analysis of performance and cost factors in searching large text databases using parallel search systems, Journal of American Society of Information Science 45 (7) (1994) 443–464.
[25] D. Hawking, Scalable text retrieval for large digital libraries, in: Proceedings of the First European Conference on Research and Advanced Technology for Digital Libraries, Pisa, Italy, September 1997.
[26] G. Trent, M. Sake, "Webstone: the first generation in HTTP server benchmarking, Technical report, MTS, Silicon Graphics Inc., February 1995.
[27] G. Banga, P. Druschel, Measuring the capacity of a Web server, Proceedings of USENIX Symposium on Internet Technology and Systems, California, USA, December 1997.
[28] J. Almeida, P. Cao, Measuring Proxy performance with the Wisconsin Proxy Benchmark, Technical document, Department of Computer Science, University of Wisconsin-Madison, USA, July 1997.
[29] IEEE 1355 DS link technology, available at: http://hsi.web.cern.ch/hsl/dshs/publications/rt97/html/rt97.html.

Charles Iheagwara is currently employed as an Information Technology Security Engineer at Edgar Online, Inc. in Rockville, Maryland, USA. He is licensed as a Professional Engineer (PE) and has many of the industry's top certifications including Cisco's CCNP and Microsoft's MCSE. He has completed the requirements for the Ph.D. degree in Computer Science at the School of Computing, University of Glamorgan, Wales, UK, where he specializes in Intrusion Detection Systems.

Andrew Blyth received his Ph.D. in 1995 from the University of Newcastle, UK. He is currently employed as a senior lecture at the School of Computing, University of Glamorgan, UK, where he specializes in Research in the areas of Information Warfare and Intrusion Detection Systems.

Appendix 4

"Evaluation of the Performance of ID Systems in a Switched and Distributed Environment: The RealSecure Case Study." Computer Networks Journal Vol 39 -2

passive surveillance mechanisms to monitor network traffic for signs of malicious or anomalous (e.g., potentially erroneous) activity. Such tools attempt to provide network administrators timely insight into noteworthy exceptional activity. Real-time monitoring promises an added dimension of control and insight into the flow of traffic between the internal network and its external environment. The insight gained through fielded network traffic monitors could also aid sites in enhancing the effectiveness of their firewall filtering rules.

An intrusion detection (ID) system is a tool that attempts to perform intrusion detection. Different ID systems have differing classifications of "intrusion"; a system attempting to detect attacks against web servers might consider only malicious HTTP requests, while a system intended to monitor dynamic routing protocols might only consider RIP spoofing. Regardless, all ID systems share a general definition of "intrusion" as an unauthorized usage of or misuse of a computer system.

Typically, intrusions take advantage of system vulnerabilities attributed to misconfigured systems, poorly engineered software, mismanaged systems, user neglect or to basic design flaw in for instance some Internet protocols. An ID system is a fast moving market with new players entering continuously. Commercial tools range from the widely available anti-viruses, to enterprise tools (e.g., Cisco/Netranger), to NT centric (e.g., Internet Security Services/RealSecure) and to configurable freeware (e.g., Network Flight Recorder). In fact such tools only detect suspicious events and report the intrusion and/or attempt to the operator. They do not (yet) include decision-making support for preventive or recovery actions once.

Generally, ID system are classified as mechanisms for parsing and filtering hostile external network traffic [1,2] that could reach internal network services and they have become widely accepted as prerequisites for limiting the exposure of internal network assets while maintaining interconnectivity with external networks. The encoding of filtering rules for packet- or transport-layer communication should be enforced at entry points between internal networks and external traffic. Developing filtering rules that strike an optimal balance between the restrictiveness necessary to suppress the entry of unwanted traffic, while allowing the necessary flows demanded for user functionality, can be a non-trivial exercise [3].

ID as an important component of a security system, complements other security technologies. By providing information to site administration, ID system allows not only for the detection of attacks explicitly addressed by other security components (such as firewalls and service wrappers), but also attempts to provide notification of new attacks unforeseen by other components. ID systems also provide forensic information that potentially allows organizations to discover the origins of an attack. In this manner, ID systems attempt to make attackers more accountable for their actions, and, to some extent, act as a deterrent to future attacks.

At its most fundamental level, ID system is a collection of detection modules also called sensors with unique attack recognition and response capabilities. Two classes are discernable:

- *Network sensors*: that monitor the raw, unfiltered traffic on enterprise networks, looking for patterns, protocol violations, and repeated access attempts that indicate malicious intent.
- *OS sensors*: These sensors perform real-time intrusion monitoring, detection, and prevention of malicious activity by analyzing kernel-level events and host logs.

The detection modules are deployed at strategic locations across the enterprise network in order to stop attacks, misuse, and security policy violations before damage is done. When an ID system detects unauthorized activity, it can respond in a number of ways, automatically recording the date, time, source, and target of the event, recording the content of the attack, notifying the network administrator, reconfiguring a firewall or router, suspending a user account, or terminating the attack.

Because of its importance within a security system, it is critical that ID systems function as expected by the organizations deploying them. In order to be useful, site administration needs to be able to rely on the information provided by the

system; flawed systems not only provide less information, but also a dangerously false sense of security. Moreover, the forensic value of information from faulty systems is not only negated, but potentially misleading.

Due to the implications of the failure of an ID component, it is reasonable to assume that the performance of ID systems are themselves crucial to an organization's security as they could become logical targets for attack. A smart intruder who realizes that an ID system has been deployed on a network she is attacking will likely attack the ID system first, disabling it or forcing it to provide false information (distracting security personnel from the actual attack in progress, or framing someone else for the attack).

As with any other technology, there are pitfalls in the current implementation of commercially available IDS. The pitfalls include the issues of variant signatures, false positive and negative alerts, data overload, difficulties to function effectively in switched environments and scale up issues.

This paper is intended to address one of the (difficulties to function effectively in switched environments) issues mentioned above. Thus, in order to gauge the ability of currently available IDS to effectively function in switched and distributed environment, the goal of the research in this paper is therefore:

1. To provide an evaluation of the performance of IDS in a switched and distributed environment; and
2. To analyze the impact of the characteristics associated with traffic flow on the performance of the IDS.

Because of the importance of surveillance on network traffic, ID systems have been studied in a wide variety of areas under different contexts. The following section provides an overview of previous studies.

2. Related work

The increasing use of E-commerce in the last couple of years has given impetus to the rise and growth of implementations of various secur systems to contain the rising waves of netwo attacks which comes in different forms and shad including unwanted intrusion into corporate I tranets. One of such mechanisms is ID syste which is used to detect and in some cases det attacks. Different technologies of these systen have been developed and it will be appropriate state that network ID systems are driven off interpretation of raw network traffic. They attem to detect attacks by watching for patterns of su picious activity in this traffic. Network ID systen are good at discerning attacks that involve lov level manipulation of the network, and can easil correlate attacks against multiple machines on network.

The significance of ID system has become ver much pronounced in complex network architec tures which often are inundated with a mesh o packet forwarding and routing devices known a switches and routers. Such networks are known a switched and distributed.

In a distributed and switched environment, th most obvious aspect of ID system to attack is it accuracy. The accuracy of ID system is compromised when something occurs that causes the system to incorrectly identify an intrusion when none has occurred (a "false positive" output), or when something occurs that causes the ID system to incorrectly fail to identify an intrusion when one has in fact occurred (a "false negative").

Research into and development of automated ID systems has been under way for well over 12 years. By now a great number of systems have been deployed in the commercial or government arenas, but all are limited in what they do. The creativity of attackers and the ever-changing nature of the overall threat to targeted systems have contributed to the difficulty in effectively identifying intrusions. While the complexities of host computers are already making intrusion detection a difficult task, the increasing prevalence of distributed network-based systems and insecure networks such as the Internet has greatly increased the need for ID.

Previous and present ID system research that relate to the technological approach of ID systems, are identified into three categories:

(i) modeling—misuse or anomaly detection;
(ii) analysis; and
(iii) deployment.

Detection is performed in the misuse detection model by looking for specific patterns or sequences of events representing previous intrusions (i.e., looking for the "signature" of the intrusion. It is a knowledge-based technique and only known intrusions can be detected. This is the more traditional ID technique, which is usually applied, in for instance the anti-virus tools.

In the anomaly detection model, this is realized by detecting changes in the patterns of utilization or behavior of the system performs detection. Building a model that contains metrics derived from normal system operation and flagging as intrusive any observed metrics that have a significant statistical deviation from the model perform it. The approach is behavior-based and should be able to detect previously unknown intrusions. It is a research and development (R&D) area in which currently innovative modeling paradigms are explored which is inspired from biological systems. Pioneers in this area are from the University of New Mexico whose work is based on the idea that ID systems should be designed to function like the way the human natural immune system distinguishes between "self" from "non-self" antibodies.

The main challenge with this approach, like for every behavior-based technique, is to model the "normal" behavior of a process. Learning the activity of the process in a real environment can do this. Another approach, advocated by IBM research, consists in describing the sequences of audit events (patterns) generated by typical UNIX processes. Another method developed by Nokia is based on Kohonen Self Organizing Maps (SOM).

Off-line vs. real-time analysis [4] is another area where more conventional classification divides ID systems into systems which operate after the event and rely on analysis of logs and audit trails for preventive action and those that attempt real-time monitoring in the hope that precursor signs of abnormal activity give indication that corrective action is possible before real damage occurs.

The work presented in this paper are extensions of earlier works on ID system and analytical methods for detecting anomalous or known intrusive activity [4–7]. In the past, emphasis has been placed on session activity within host boundaries given the fact that the primary input to ID tools, audit data, is produced by mechanisms that tend to be locally administered within a single host or domain. However, as the importance of network security has grown, so has the need to expand ID technology to address network infrastructure and services. In this research effort, we explore the extension of ID methods to the analysis of network activity under a switched and distributed architecture.

Network monitoring, in the context of fault detection and diagnosis for computer network and telecommunication environments, has been studied extensively by the network management and alarm correlation community [8–11]. The high-volume distributed event correlation technology promoted in some projects provides an excellent foundation for building truly scalable network-aware surveillance technology for misuse. However, these efforts focus primarily on the health and status (fault detection and/or diagnosis) or performance of the target network, and do not cover the detection of intentionally abusive traffic in distributed and switched environments. Indeed, some simplifications in the fault analysis and diagnosis community do not translate well to a malicious environment for detecting intrusions. Examples include assumptions of stateless correlation, which precludes event ordering; simplistic time-out metrics for resetting the tracking of problems; ignoring individuals/sources responsible for exceptional activity.

The scale of scientific research of ID systems has grown by leaps and bounds in the last couple of years. Studies of ID systems attempting to address the issue of network surveillance include the Network Security Monitor developed at UC Davis [12], and the Network Anomaly Detection and Intrusion Reporter [13] developed at Los Alamos National Laboratory. Both performed broadcast LAN packet monitoring to analyze traffic patterns for known hostile or anomalous activity. Further, research by UC Davis in the Distributed ID system [14] and later Graph-based ID system [15] projects has attempted to extend intrusion-monitoring

capabilities beyond LAN analysis, to provide multi-LAN and very large-scale network coverage.

Network Traffic Intensity measurement has been investigated by Morris [16] and Maimon [17]. Intensity measures distinguish whether a given volume of traffic appears consistent with historical observations. These measures reflect the intensity of the event stream (number of events per unit time) over time intervals that are tunable. Alternatively, a sharp increase in events viewed across longer durations may provide insight into a consistent effort to limit or prevent successful traffic flow. Morris [16] investigated intensity measures of transport-layer connection requests, such as a volume analysis of SYN-RST messages, which could indicate the occurrence of a SYN-attack against port availability (or possibly for port scanning). Maimon [17] explored intensity measures of TCP/FIN messages as a variant considered to be a more stealthy form of port scanning.

In their studies [16,17], the authors contend that monitoring overall traffic volume and bursty events by using both intensity and continuous measures provides some interesting advantages over other monitoring approaches, such as user-definable heuristic rules that specify fixed thresholds. In particular, the intensity of events over duration is relative in the sense that the term "high volume" may reasonably be considered different at midnight than at 11:00 a.m. The notion of high bursts of events might similarly be unique to the role of the target system in the Intranet (e.g., web server host versus a user workstation).

Traffic analysis with signature analysis has been studied [2,6,18–21]. Signature analysis is a process whereby an event stream is mapped against abstract representations of event sequences known to indicate the target activity of interest. Determining whether a given event sequence is indicative of an attack may be a function of the preconditions under which the event sequence is performed.

The use of coding schemes for representing operating system penetrations through audit trail analysis was also the focus of other research works [6,18,19]. Using basic signature-analysis concepts, it was shown that some detection methods could support a variety of analyses involving packet and transport datagrams as event streams. For example, address spoofing, tunneling, source routing [20], SATAN [21] attack detection, and abuse of ICMP messages (Redirect and Destination Unreachable messages) [2] could all be encoded and detected by signature engines that guard network gateways.

The advent of large-scale commercial ID systems tend to have given a relative assurance to the information technology community that has been very anxious to maximize the use of these highly advertised ID systems as added armor to secure network systems. Many IDS products have been deployed in commercial and corporate networks. With this has come a shift in research focus in so many areas. One of such is in the area of the IDS performance.

Richards [22] evaluated the functional and performance capabilities of the industries leading commercial type ID system. In the areas tested, the performance of the ID system was rated based on their distinctive features, which were characterized into different performance indexes. The research work represented a new direction for ID systems in that it moved the focus away from scientific concepts research to performance evaluation of the industries best products. However, the study was limited to a small proto design isolated and non-switched network which did not reveal the impact of packet switching on the accuracy and ability to capture attack packets in their entirety. We believe that an effectiveness measurement study must take into account the complexity that characterize the existence of actual network traffic pattern and its logical effect on ID system study.

In our research, we leveraged the work of Richards [22] to an actual network of distributed and switched topology. Our IDS evaluation studies treat the relationship between deployment techniques and attack system variables and the performance of the IDS.

Porras and Valdes [23] discussed ID system failures in terms of deficiencies in accuracy and completeness, where accuracy reflects the number of false positives and completeness reflects the number of false negatives. We related our work in the context of interpretative analysis to their work.

In Section 3 we describe the evaluation of IDS performance in a switched and distributed environment.

3. Experimental work

3.1. Objective

Our research objective is to determine the performance characteristics and effectiveness of ID system using the RealSecure Suite for a distributed and switched network infrastructure.

3.2. Framework

The framework of this study is the extension of the work of Richards [22]. This research work was conducted on a non-distributed and non-switched network. Hence factors such as packet loss due to routing and switching were not taken into account. In contrast, our study was conducted on a switched and distributed network. This is based on the fact that routing and switching constitute a major factor in network attacks as described in Section 4 of this paper thus making it proper to conduct performance evaluation study on an environment similar to the actual IDS deployment environment. It is therefore imperative that the impact of routing and switching be taken into account when gauging the effectiveness of ID system in a distributed environment. Thus, the deployment of the ID system in the switched and routed network was intended to determine by how much the performance of the ID system sensor is impaired by packet switching and other network conditions.

3.3. Baseline and evaluation criteria

The basic performance indicators of any IDS should be reflected in the success or failure of event analysis, which are quantitatively measured for qualities such as accuracy and performance that are assessable through testing. A more difficult but equally important metric to assess is completeness. With regard to network monitoring, inaccuracy is reflected in the number of legitimate transactions flagged as abnormal or malicious (false positives), incompleteness is reflected in the number of harmful transactions that escape detection (false negatives), and performance is measured by the rate at which transactions can be processed.

Equally, for an IDS evaluation, the standard of measurement is the ability of the IDS to satisfy the design, deployment functionality and performance requirements described in Appendix A. The RealSecure IDS uses the pattern matching technique. The theory behind this is that a pattern-matching system does not know the contents of the packets, and must match packets for different patterns. A pattern-matching system looks for patterns on ranges of ports where the exploit program typically run.

Within the limits of our experiment, the evaluation criteria used is the percentage of attacks captured by the IDS against the tunable experimental parameters i.e., throughput, monitoring technique and attack signatures.

The following are the characteristics of the attack sets used in the experiment.

Attack taxonomy—intrusion attacks have been presented in the scheme of Kumar [26] and can be represented by an event or series of events. It is the relationship of these events to one another that provides the basis for recognizing differing attack types. The class hierarchy is shown in Fig. 1. Under the attack taxonomy, intrusions fall into two categories: namely misuse and anomalous behavior. Misuse comprises attacks that are already known and whose behavior can be specified while anomalous behavior describes attacks involving unusual use of the system resources.

The manifestations of the misuse attack types can be grouped into the modes shown in Table 1. In our experiments the attack set described below falls under the active misuse attack type.

Header attack—the purpose is to gauge the ability of the ID system to handle IP packet header attacks. The *LAND attack* is a typical kind of this attack in which a SYN packet is sent with the same source and destination IP address and port. This forces the IP stack into a progressive loop that crashes the stack.

Reassembly attack—the purpose is to gauge the ability of the ID system to reassemble fragmented IP fragments and identify attacks that occur over

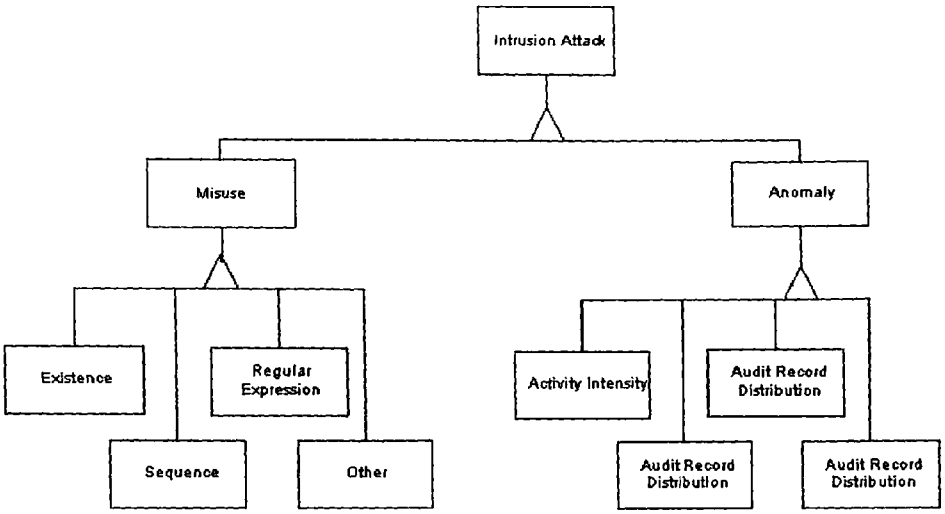


Fig. 1. The class hierarchy of Intrusion attacks. (The forks below the rectangles represent inheritance.)

multiple packets. There are two types of attacks associated here:

- The *TearDrop* attack that is initiated by sending multiple fragmented IP packets, that when reassembled, have data portions of the packet that overlap. This causes protocol and/or the system to become unstable;
- The *Ping of Death* attack is initiated by sending multiple fragmented ICMP packets, which when reassembled, have a data portion of greater than 65,535 bytes. As this is a violation of the TCP/IP specification, it causes the TCP/IP stack to crash on vulnerable computers.

Empty packet attack—the goal is to quantify the ID system's ability to capture each packet without experiencing packet loss (throughput). There was no attack initiated, and the ID system sensor was configured without any signature loaded. We performed the empty packet attack to characterize this.

3.4. Test bed

3.4.1. Attacker

The attacker is an Intel-based system running Windows NT 4.0 server (with Service Pack 5) loaded with Network Associates CyberCop scanner located outside the perimeter of the Intranet.

3.4.2. Target

The targets are two Windows NT 4.0 servers dispersed at the following network locations depicted in Fig. 2:

1. News server with the IP address of 216.133.249.4.
2. Web server with IP address of 192.168.233.2.

3.4.3. Load generator

The load generator was a Windows NT 4.0 server running Shomiti Surveyor 2.4. located outside the perimeter of the Intranet.

3.4.4. RealSecure sensor

It was a standard PC with the following hard and software configuration: Pentium II 300 MHz processor, 128 MB RAM, 100 MB disk space plus 100 MB per managed sensor on the console. NT 4.0 Workstation with SP5, 100 Mbps Ethernet adapters. The sensor analyzes the packets on the wire and alerts if it senses an attack.

Table 1
The different modes of the misuse intrusion attack type

Mode of misuse	Description
External misuse	
Visual spying	Observation of keystrokes or screen
Misrepresentation	Deceiving operators and users
Physical scavenging	Dumpster diving for printouts, floppy disks, etc.
Hardware misuse	
Logical scavenging	Examining discarded/stolen media
Eavesdropping	Intercepting electronic or other information
Interference	Jamming, electronic or otherwise
Physical attack	Damaging or modifying equipment or power
Physical removal	Removing equipment and storage media
Masquerading	
Impersonation	Using false identities external to the computer system
Piggybacking attacks	Usurping communication lines and workstations
Spoofing attacks	Using playback, creating bogus nodes and systems
Network weaving	Masking physical whereabouts or routing
Pest programs	
Trojan Horse attacks	Implanting malicious code, sending letter bombs
Logic bombs	Setting up time or event bombs
Malevolent worms	Acquiring distributed resources
Virus attacks	Attaching to programs and replicating
Bypasses	
Trapdoor attacks	Utilizing existing flaws in the system and misconfigured network programs
Authorisation attacks	Password cracking etc.
Active misuse	
Basic active attack	Creating, modifying, entering false or misleading information
Incremental attack	Using salami attacks
Denial of service	Perpetrating saturation attacks
Passive misuse	
Browsing	Making random and selective searches
Inference, aggregation	Exploiting database inferences and traffic analysis
Covert channels	Exploiting covert channels or other information leakage
Inactive misuse	Wilfully failing to perform expected duties, or committing errors of omission
Indirect misuse	Preparing for subsequent misuses, as in off-line pre-encryption matching, factoring large numbers to obtain private keys, auto-dialer scanning

3.4.5. Attack signatures

We used the standard (5.0) attack detector policy. Each signature was set to count the number of packets it triggers. The signatures for all attacks were enabled with alert console and the log to database responses was also enabled. Other attack signatures not needed in the attack were disabled and the RealSecure kill response was not used as the span ports were configured only for uni-directional traffic.

3.4.6. Deployment technique

In order to fully evaluate the impact of location and deployment techniques on the ID system performance we conducted the tests in the following deployment topologies (Table 2).

1. Outside decoy: to detect all traffic coming into the network from the Internet, the RealSecure sensor was plugged into the Century Tap placed between the router/switch and the applicable 100BaseT LinkSwitch on the given network, or plugged into the management port of the LinkSwitch 3000, when the mirroring technique is used (Fig. 3).
2. Inside decoy: the RealSecure sensor was plugged into the Century Tap between the router/switch and the applicable 100BaseT LinkSwitch on the given network (as shown in Fig. 4 for the Web server attack), or plugged into the management port of the applicable LinkSwitch when the mirroring technique is used.

When using the port mirroring technique, we plugged the RealSecure sensor directly into the management port of the switch into which the other traffic ports were spanned. The management port mirrors all the traffic coming through the ports that are spanned. The RS sensor was configured to operate in a stealth mode i.e., with two

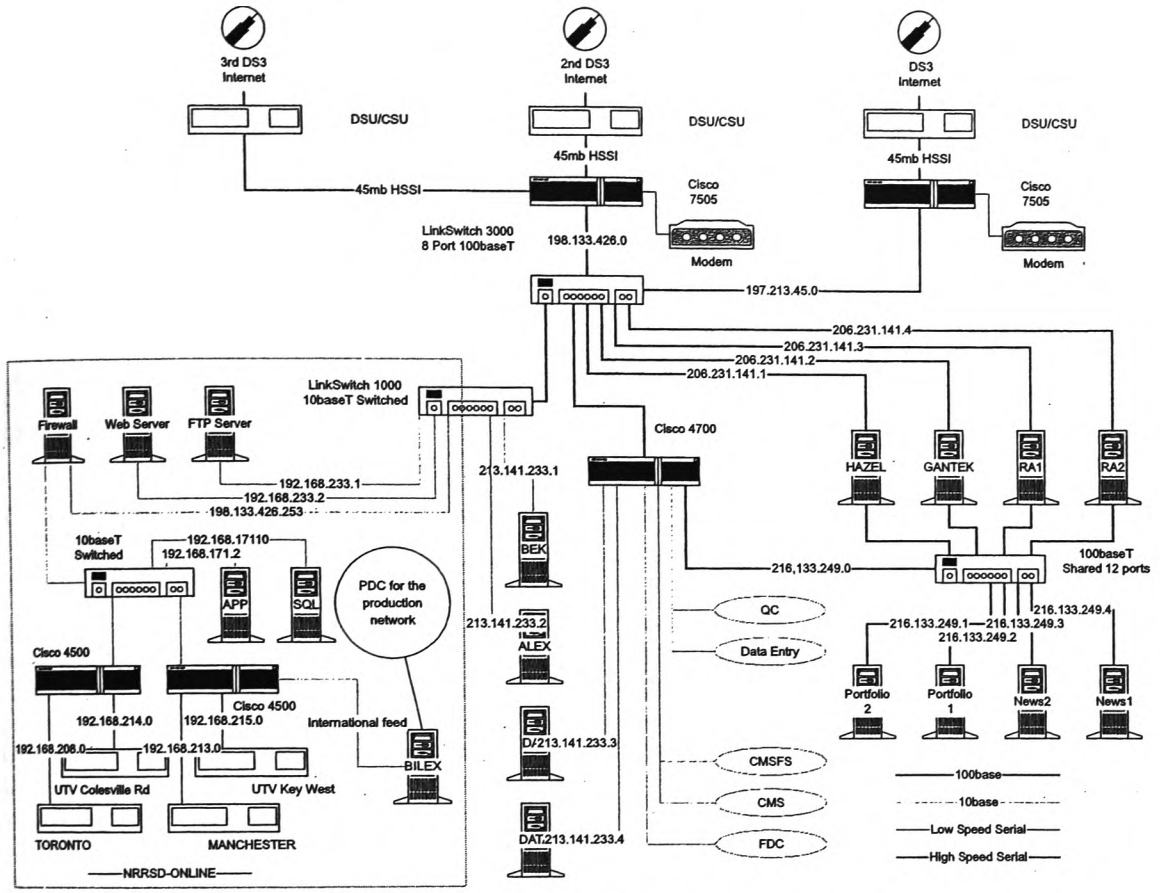


Fig. 2. Setup and topology of the network test bed.

Table 2
Topological locations of the Century Tap and RealSecure sensor

Decoy	Location
Outside decoy	Between the Cisco 7505 router and LinkSwitch 3000 on the 198.133.426.0 network
Inside decoy for Web server attack	Between the LinkSwitch 3000 and LinkSwitch 1000 (Fig. 4)
Inside decoy for News2 server attack	Between the Cisco 4700 router and 100BaseT 12 ports shared Switch on the 216.138.240.0 network

NICs. In this case, one card operates in a promiscuous mode which implies that it will not be bound to the TCP/IP stack thus will have no IP address, and will do the actual monitoring. This is the interface that is plugged into the switch. The other NIC was bound to the TCP/IP stack and have an IP address that is used to communicate back to the console, send email notifications, and send SNMP traps for triggered events.

3.5. Methodology

Our methodology takes a pragmatic approach towards the issue of quantitative and qualitative treatment of intrusion detection analysis using the RealSecure ID system techniques in a switched and distributed network. We begin by describing the setup of the test bed, deployment of the ID system sensors, attack systems and targets.

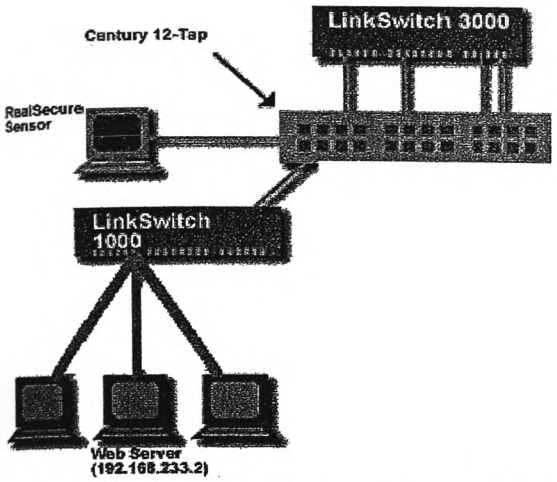


Fig. 3. Deployment of the RealSecure sensor into the Century Tap for the outside decoy attack set.

Further, we define the attack sets. The initiation, sequence, generation and launching of the attack was then described in the test procedure (Section 3.6). Based on the test results, we give a qualitative analysis of contributing factors for inaccuracies obtained on the test results. We describe the analytical techniques that are pertinent to ID system, and discuss their relevance in analyzing malicious, faulty, and other exceptional network activity.

We injected specifically configured attack packets onto a test network, on which the subject RealSecure (ID) system was running. By tracking the subject's monitoring console output, we were able to observe the detection performance and functional characteristics of the ID system and the system's underlying TCP/IP implementation. In all cases, our tests involved interactions between injected packets and the host target of attack. In each test, this target host was the explicit addressee of all the attack packets. With this, the presence of the target host allowed us to easily create the needed TCP connections for the subject ID system to monitor. In addition, the target host also acted as a control for our experiments. The target's response to injected attack packets allowed us to empirically record the behavior of the performance of the ID system in all test categories, and contrast the observed performance with the theoretical design performance of the ID system.

The actual tests were preceded by a series of baseline tests against the target host to ensure that the subject ID system was configured and functioning properly according to design specifications. In almost all test cases, a process on the target host ran which accepted incoming TCP connections on the HTTP port and printed any input obtained from the machine's TCP stack. By examining the output of this process, we were able to deduce whether the subject ID system should have detected the attack based on the network conditions we created.

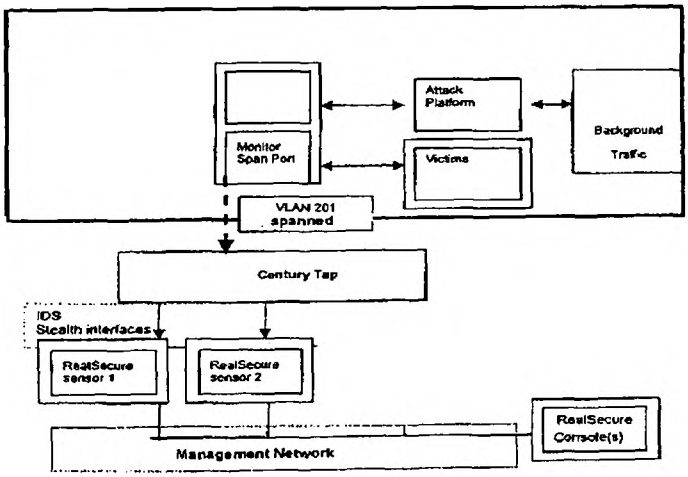


Fig. 4. Test environment logical diagram.

We mimicked a series of defined and designated attack sets that are incorporated into the software suite of the ID system. Each attack exploits a specific attack signature in the areas tested. In each test, we created the necessary network traffic flow and monitoring conditions that relied on certain differences in deployment and monitoring technique in order to evaluate their full impact on the performance and functionality index of the ID system. In each test, the specific packets injected into the network differed subtly. The subject ID system reacted to each test by capturing all, partially capturing or not capturing the attack. By considering the IDS' output and the specific types of packets used for the test, we were able to deduce the significant characteristics of the IDS.

Within the context of a hierarchical model for a distributed architecture, RealSecure's capability to monitor entry points that separate external network traffic from an enterprise network and its constituent local domains is evaluated and compared with its capability to monitor network traffic inside the decoy. We present these monitoring techniques in the context of their effectiveness.

We define and consider the characteristics of the candidate attack streams that pass through network entry points. Critical to the effective ID system detection of attacks is the careful selection and organization of these event streams such that an analysis based on a selected event stream will provide meaningful insight into the target activity. We identify effective techniques for deploying the ID system sensor given specific test objectives. We explore the impact of ID system anomaly detection and how traffic flow analysis can be applied to identify activity worthy of review and possible response. More broadly, we discuss the correlation of analysis of network traffic and the results produced in our surveillance components deployed in the entry points and specific inside locations of our protected Intranet. We discuss how events of limited significance to a local surveillance monitor may be aggregated with results from other strategically deployed monitors to provide insight into more wide-scale problems or threats against the Intranet.

3.6. Test procedure

The test suite assesses the sensor's packet processing capability at both low and high network load. The Shomiti Surveyor (version 2.4) generated the network load and the attack was captured with Network Associates Sniffer Pro and launched at wire speed or as close to wire speed as the sniffer could send it. Network attack traffic load on the 100 Mbps network was generated in the following incremental levels: 1000 packets; 5000 packets; 10,000 packets; 12,000 packets; 25,000 packets; and 45,000 packets per second. All packets were 64 bytes in size. For each attack set the sensor was configured with the appropriate signature.

Within the experimental setup environment, multiple instances of each attack sets were launched at the targets. The tests were conducted in the following sequence:

1. generation of attack packets,
2. generation of background traffic and verification with a network analyzer,
3. launching of attacks against target systems,
4. records and auditing of attack detection performance of the RealSecure sensor,
5. termination of traffic generation,
6. purging (clearing) of the console display,
7. repetition of Steps 3–6, for each utilization two more times for a total of five trials,
8. repetition of Steps 1–7 with a single sensor at specified background utilizations.

3.7. Experimental results

We have presented the results obtained in the tests in Tables 3–9.

In Table 3, the data in each cell represents the number of attacks the sensor detected out of a potential 1000 attacks launched at the Web server. Figs. 5 and 6 are the graphical representations.

In Tables 4–9, the numbers in the first row corresponds to the number of packets in each instance attack whereas the number in each cell is the average number of attack packets from five tests the sensor detected in each instance attack set at 40% network utilization.

Table 3
Percentage of attacks captured at 20% and 40% network utilizations

Monitoring technique	Inside decoy with Tap		Inside decoy with port mirroring		Outside decoy with Tap		Outside decoy with port mirroring	
No. of attacks	1000		1000		1000		1000	
% Utilization	20	40	20	40	20	40	20	40
Packet capture attack	824	733	817	673	893	827	879	727
Land attack	807	625	789	647	861	695	853	711
Ping of death attack	815	613	797	601	868	653	858	718

Table 4
Inside decoy Century Tap deployment tests results for the Web server directed attack

Attack type	1000	5000	12,000	25,000	45,000
Empty packet	796	4313	9300	15,441	30,922
LAND	850	4390	9733	16,835	32,777
Ping of death	818	4341	9682	16,411	31,340

Table 5
Outside decoy Century Tap deployment tests results for the Web server directed attack

Attack type	1000	5000	12,000	25,000	45,000
Empty packet	893	4845	10,450	17,347	34,741
LAND	949	4932	10,935	18,916	36,828
Ping of death	919	4878	10,876	18,439	35,213

Table 6
Inside decoy port mirroring tests results for the Web server directed attack

Attack type	1000	5000	12,000	25,000	45,000
Empty packet	606	3440	7244	11,352	17,790
LAND	668	3562	7401	12,739	24,221
Ping of death	668	3558	7463	13,314	25,634

Table 7
Inside decoy port mirroring tests results for the News2 server directed attack

Attack type	1000	5000	12,000	25,000	45,000
Empty packet	665	3537	7630	12,660	25,362
LAND	693	3600	7984	13,808	26,884
Ping of death	671	3569	7938	13,460	25,707

Table 8
Inside decoy Century Tap deployment tests results for the News2 server directed attack

Attack type	1000	5000	12,000	25,000	45,000
Empty packet	740	4205	8830	13,840	21,698
LAND	814	4335	9007	15,531	29,520
Ping of death	814	4339	9099	16,233	31,252

Table 9
Outside decoy port mirroring tests results for the Web server directed attack

Attack type	1000	5000	12,000	25,000	45,000
Empty packet	830	4718	9920	15,550	24,375
LAND	915	4871	10,120	17,450	33,168
Ping of death	915	4875	10,224	18,239	35,115

4. Analysis

An analysis of the test results comparative to the metric design values showed a fairly acceptable result especially with the Century Cap deployment technique for the packet header (land) attack, and the IP fragment reassemble (Ping of Death) attack. The results could not just be explained by drawing contrasts with design parameters, but also by analyzing the impact of other intrinsic factors of the test network that constitute the most serious challenge to the effective implementing RealSecure in a switched environment.

Attack detection is a crucial guage of the effective performance of ID system because, if the ID system sensor has difficulty capturing, reassembling, or identifying attacks, the attacks will go unnoticed—thus defeating the purpose of having an ID system. Characterizing the detection ability of the ID system is very important for production networks that are generally very busy, where the major challenge is determining how much of the attack traffic the sensor can handle before its performance begins to degrade as it drops packets.

The test results, which are graphically represented in Figs. 5–12, established obvious trends that could be a useful guide for network security managers who are charged with the responsibility of implementing network security. In analyzing the test results it is worth noting that there are many factors that could adversely affect the effectiveness of the IDS some of which are discussed below:

1. deployment (monitoring) technique i.e. port mirroring or the use of packet loss restricting devices;
2. network traffic load and bandwidth;

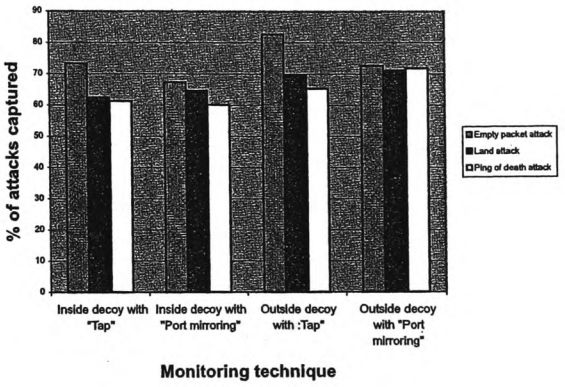


Fig. 5. Percentage of attacks captured at 40% network utilization.

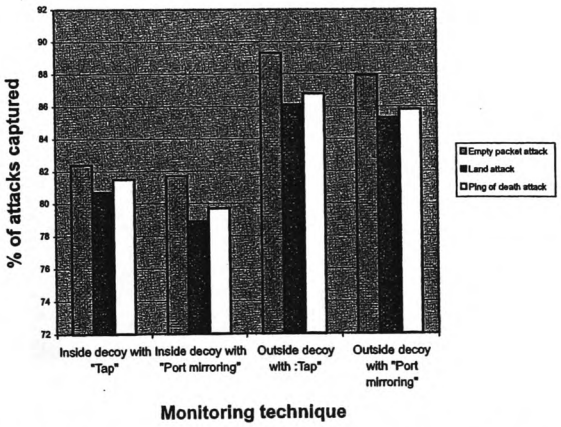


Fig. 6. Percentage of attacks captured at 20% network utilization.

3. location of the sensor(s);
4. the condition of network packet forwarding devices; and
5. network condition.

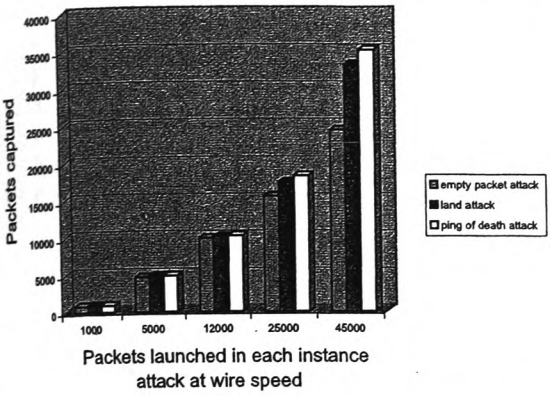


Fig. 7. Outside decoy port mirroring tests results for the Web server directed attack.

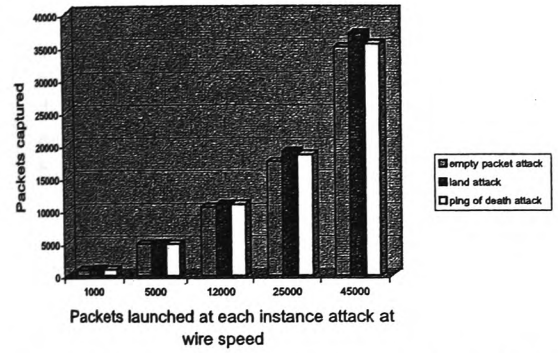


Fig. 8. Inside decoy port mirroring tests results for the News2 server directed attack.

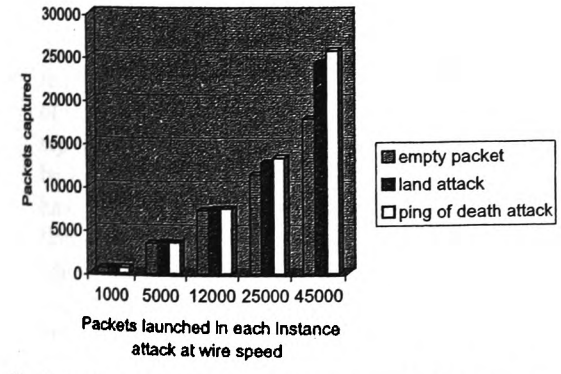


Fig. 9. Inside decoy port mirroring tests results for the Web server directed attack.

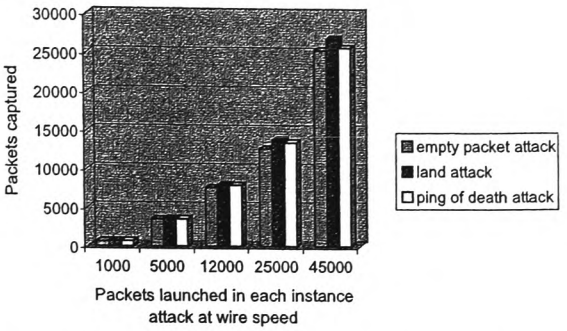


Fig. 10. Inside decoy port mirroring tests results for the News2 server directed attack.

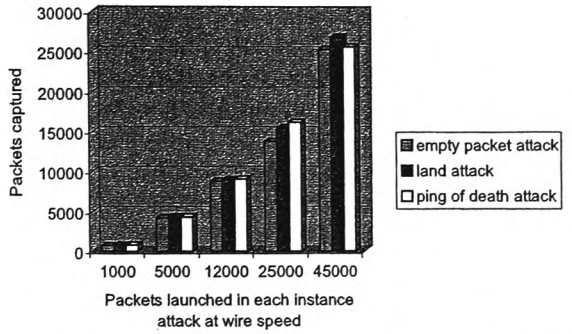


Fig. 11. Inside decoy Century Tap deployment tests results for the Web server directed attack.

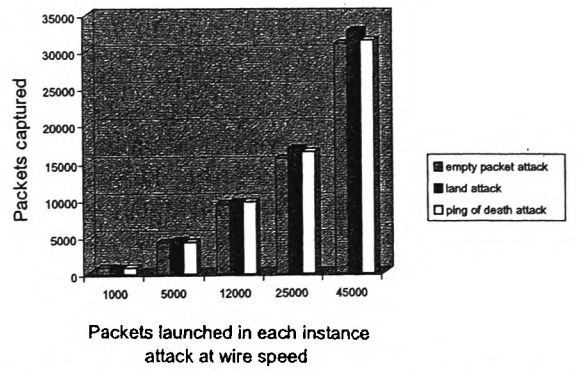


Fig. 12. Inside decoy Century Tap deployment tests results for the News2 server directed attack.

4.1. Monitoring techniques

Network capture and analysis in a switched LAN environment usually means "tapping" the

switch's lines by using a "mirror" port or deployment in other tapping configurations. In this approach, traffic is copied from one "source" port to another destination or "mirror" port; an analyzer attached to the mirror port then sees all of the traffic entering and exiting the source port—in theory.

With regard to port mirroring, a problem can arise when the switch ports are configured for full duplex operation. Since full duplex allows traffic to flow simultaneously in both directions, it effectively doubles the available network bandwidth. A mirror port on a switch can monitor traffic in one direction but not two (i.e., it can copy traffic from the source port but cannot monitor traffic to it), and is therefore a logical half duplex operation. Even if the switch copies traffic from both transmit and receive channels on the source port, the traffic will eventually be forced onto the transmit channel of the mirror port. For this reason, mirroring a full duplex source port may cause packet loss as traffic on the full duplex source port exceeds the available bandwidth of the mirror port.

In contrast, a tapping device like the "Century Tap" eliminates the problems associated with port mirroring such as switch performance degradation, inability to mirror errors (packet undersize or oversize, and packets with bad CRC), inability to view VLAN traffic on some switches and inability to show both sides of the full duplex link.

4.2. Network throughput

Over utilization could have a negative impact on the ability of the sensor to capture attacks. At times, there could be inaccurate report on the level of network utilization. For instance, the utilization reported by the analyzer on the mirror port could be less than 100% when in fact, not all packets have arrived safely. That is because over utilization can occur at intervals shorter than that over which the analyzer is reporting its results. In other words, if actual throughput at the source port exceeds the mirror port's bandwidth for a half second, then drops significantly, the analyzer may never see throughput approaching 100% in a given one-second sample.

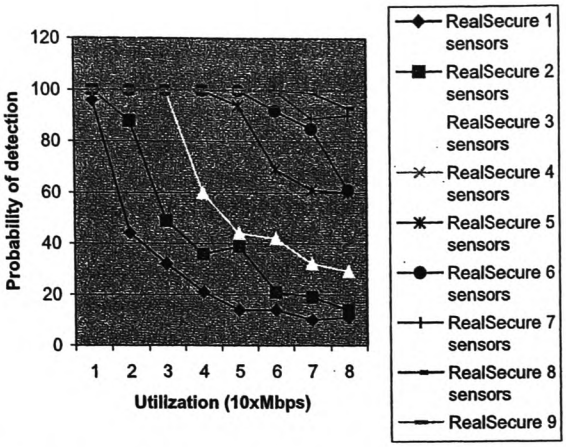


Fig. 13. Probability of detection vs. percent utilization.

Recent evaluation study [27] by Internet Security systems and Top Layer Networks on the probability of detection by the RealSecure sensor shown in Fig. 13 established that with increase in network utilization, the probability of detection falls, but performance is enhanced with the addition of additional sensors.

4.3. Deployment (host or network) locations

Our test results show that the deployment location of the ID system sensor plays an important role on its performance ability.

Intrusion monitoring can either be sited at the computer system that is the putative target or placed on a network level where traffic can be evaluated or where information aggregated from various hosts can give insight in co-ordinated attack scenarios.

The effect of deployment on the performance of the ID system sensor has been presented in the EMERALD concept [24]. The distributed framework concept (EMERALD module) illustrated in Fig. 14 depicts an example enterprise network consisting of interconnected local network domains.

Inside the perimeter of the enterprise, each local domain maintains its traffic filtering control (F-boxes) over its own sub networks. These filters enforce domain-specific restriction over issues such

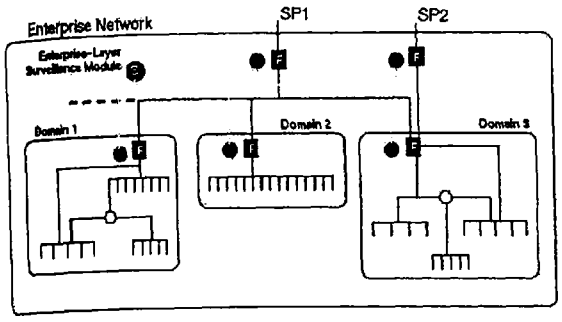


Fig. 14. The EMERALD module for the deployment of surveillance in an enterprise network.

as UDP port availability, as well as acceptable protocol traffic. The surveillance monitors are represented by the S-circles, and are deployed to the various entry points of the enterprise and domains. The surveillance modules develop analysis results that are then directed up to an enterprise-layer monitor, which correlates the distributed results into a meta-event stream.

This concept has given much weight to the correlation trend analysis in which attacks repeated against the same network service across multiple domains can also be detected through enterprise-layer correlation. For example, multiple ID system sensors deployed to various local domains in the network might begin to report, in series, suspicious activity observed within sessions employing the same network service. Such reports could lead to enterprise-layer responses or warnings to other domains that have not yet experienced or reported the session anomalies. In this sense, results correlation enables the detection of spreading attacks against a common service, which first raise alarms in one domain, and gradually spread domain by domain to affect operations across the enterprise. This in our view should provide a beneficial value towards the effective deployment of ID system.

4.4. Network conditions

Network ID systems work by predicting the behavior of networked machines based on the packets they exchange [25]. The problem with this is that a passive network monitor cannot accurately predict whether a given machine on the network is even going to see a packet, let alone process it in the expected manner. The existence of a number of factors could make the actual meaning of a packet captured by ID system ambiguous. These can be considered in the following context:

(1) A network ID system is typically on an entirely different machine from the systems it's watching. Often, the ID system is at a completely different point on the network. The basic problem facing a network ID system is that these differences cause inconsistencies between the ID system and the machines it watches. Some of these discrepancies are the results of basic physical differences, others stem from different network driver implementations. For example, consider an ID system and an end-system located at different places on a network. The two systems will receive any given packet at different points in time. This difference in time is important; during the lag, something can happen on the end-system that might prevent it from accepting the packet. The ID system, however, has already processed the packet thinking that it will be dealt with normally at the end-system.

(2) IP packet with a bad UDP checksum will not accepted by most operating systems. Some older systems might. The ID system needs to know whether every system it watches will accept such a packet, or it can end up with an inaccurate reconstruction of what happened on those machines. Some operating systems might accept a packet that is obviously bad. A poor implementation might, for example, allow an IP packet to have an incorrect checksum. If the ID system does not know this, it will discard packets that the end-system accepts, again reducing the accuracy of the system.

(3) Even if the ID system knows what operating system every machine on the network runs, it still might not be able to tell just by looking at a packet whether a given machine will accept it. A machine that runs out of memory will discard incoming packets. The ID system has no easy way to determine whether this is the case on the end-system, and thus will assume that the end-system has accepted the packet. CPU exhaustion and network saturation at the end-system can cause the same problem.

Together, all these problems result in a situation where the ID system often simply cannot determine the implications of a packet merely by examining it; it needs to know a great deal about the networking behavior of the end-systems that it's watching, as well as the traffic conditions of their network segments. Unfortunately, a network ID system does not have any simple way of informing itself about this; it obtains all its information from packet capture.

4.5. Packet forwarding/switch device

It is believed that security can be enhanced with layer two-switching technology, which sends traffic to a computer that is the target destination. The motive for this technology is performance. Generally, older technologies would broadcast each Ethernet packet to all connected computers making it possible for sensitive information to be divulged to the unintended target. The divulged information could be used to generate system attacks. The Ethernet switch added a certain amount of security by only sending traffic to its proper destination. In this case, all connected computers will not receive all traffic. They only receive broadcast traffic and traffic that was sent to them.

However, the switching technology is still not completely safe. The problem with this technology lies on how these switches handle broadcast traffic. A typical attack is to use a program that sends fake ARP requests and replies. Typically, these switches keep tables of IP addresses and Ethernet addresses. By sending in Ethernet packets with a broadcast source address, the switch may think that some or all IP addresses actually broadcast Ethernet addresses. This causes some switches to broadcast all IP traffic to all listening devices. The use of some special packet monitoring devices like the Shomiti system Century Tap, which allows visibility into 10/100 Ethernet LANs, is meant to solve the problem.

4.6. Network topology

The topology used in a test can influence the detectability of an attack path. The attack will not be detected if evasive alternative attack paths are used to launch the attack. The topology that hacker would exploit could be unsuspected net work points such as network routers that feed th target network. If these routers were owned by a ISP and not by the target network, an attacke would want to control those first few routes i order to sniff traffic, spoof DNS, and hijack net work connections and many other techniques t leverage access.

Protocol spoofing can be used to alter almo every major routing protocol. RIP and OSPF hav many attacks that can be affected simply by in jecting fictitious route information in the form c spoofed packets. Spoofing ARP packets in orde to overcome layer two-security partitioning is ar other technique to overcome topology segmenta tion.

4.7. Denial of service attack

A denial of service attack involves software tha deliberately crashes a system, or makes a system c network unusable. The attack does not involv breaking into any of the targeted servers; rather, floods them with nonsense requests and erro from Internet. These attacks are significant, a they rely upon weaknesses in the design of th Internet and TCP/IP, the set of rules (protocol for exchanging data over the Internet.

The TCP/IP protocols defines an exchange c three packets to set up a connection Fig. 15, an the first of these three packets is marked SYN, fc synchronize. In normal use, the server sends a acknowledgement to the client that is requestin

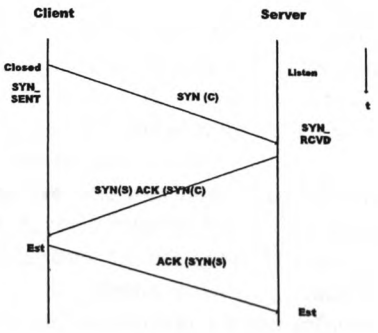


Fig. 15. TCP three way handshake, connection initiation.

the connection, and the client then responds with its own acknowledgement. When these three packets have been exchanged, both the client and the server have an established connection.

During a SYN flood attack, the target server attempts to complete the connection by sending acknowledgements to the spoofed source address. Since there is no system associated with the spoofed source address, there will either be no response, or an error response (an ICMP packet) will be generated. The TCP/IP stack that is designed to wait for a valid acknowledgement will never receive one during this attack. This will induce the TCP/IP stack to keep track of all the invalid connection attempts, and will begin ignoring valid requests.

While the SYN flood can work with relative trickle packets, there are other denials of service attacks that involve a torrent of packets. These attacks rely on overwhelming either the target system or the network leading to that system.

The task for ID systems is not just detecting the attack but also quantifying the attack. The ability of the ID system to capture the attack might be impaired by network conditions that might impair its ability to function. In this case, much will depend on if the attack is well sustained to the extent of shutting down all network services. The possibility of this depends on defensive and contingency measures in place.

5. Challenges for large-scale distributed infrastructures

ID for emerging large-scale distributed systems (e.g., global companies and virtual enterprise networks) faces a variety of difficult challenges. The most important ones can be summarized as:

Multiple attack scenarios: The anatomy of an intrusion is composed of increasingly complex attack scenarios. An attack scenario consists in a logical sequence of actions that are applied for reaching a particular strategic goal (e.g., getting confidential information). These actions are typically applied on different hosts in a network and by using a variety of tools. Moreover, a variety of different attack scenarios are possible to reach the same goal. There is a need for dynamically linking local individual events to global attack strategies in order to provide pro-active and adaptive intrusion monitoring.

Architectural approach: ID SYSTEM techniques so far concentrate on local event monitoring. Important new issues in the large-scale network context are information exchange, work division and co-ordination amongst various ID systems. An emerging architectural approach is based on autonomous local ID system agents performing event processing coupled with co-operative global problem resolution. However, the degree of autonomy of agents is subject to debate and research. Purdue University has been working on their AAFID, the Autonomous Agents for Intrusion Detection. However at the present stage, the system does not yet exploit the real mobility and autonomy aspects of agents.

Performance in complex infrastructures: Large distributed networks of systems need scalable ID system approaches for which performance is becoming an important attribute. This includes issues of timeliness of local event monitoring and communication of contextual data between nodes as well as of trust relationships between the nodes.

Challenges: Most of the individual techniques are more suitable for local event monitoring and analysis. Globally co-ordinate attack strategies require integration of methods and aggregation of disparate information sources. The critical issue lies in defining the high-level communication protocol to allow different methods of ID system to contribute to the intrusion detection process.

Integration with network management system: ID system methods must be better integrated with exiting network management systems if their widespread adoption in industry is to be guaranteed. One reason is that this should facilitate their maintenance/upgrades and a more coherent audit/log data management.

ID systems are one mechanism to respond to new business dependability/survivability needs. It is as yet unclear how to integrate ID system with other dependability mechanisms (e.g., fault tolerance, recovery mechanisms) in a wider information risk and security policy context.

6. Conclusion

We have conducted an IDS evaluation study to establish the relationship between deployment techniques, attack system variables and the performance of the IDS.

We have outlined in this paper the basic problems associated with the reliability of traffic flow, attack detection analysis, the difficulties of accuracy of attack detection and the analysis of the facts pertinent to the ID system as an effective security tool.

We have also described the attack sets and techniques used to evaluate the performance of the RealSecure ID system in a switched and distributed environment. We have presented the factors that make considerable (although some time difficult to quantify) impact on the ability of the ID system to attain optimum design performance. In the tables, we have summarized the results of our tests and have given an interpretative analysis of the results including identification of the best deployment and monitoring techniques that could enhance surveillance on production networks.

The results established that deployment of the sensor at the gateway entry (outside decoy) produced better results, and specifically:

- Performance in the port mirroring technique for the Web directed attack was better from the outside decoy than the inside decoy by 16%.
- Deployment of the sensor with the Century Tap for the News2 server directed attack had a better performance by about 16% than the same attack using the port mirroring technique.
- The performance of the sensors in the attacks to the Web and News2 servers using the port mirroring technique is identical.
- Using the Century Tap outside decoy yielded better result than the using it inside decoy by 11% for the Web based attacks.
- Equally, using the Century Tap inside decoy yielded better result than using the port mirror inside decoy by 27% for the Web based attacks.

Our studies provide justification that an effective ID system can be achieved by using a best effort delivery/deployment approach which integrates the monitoring and deployment techniqu devised in this study to maximize the benefits the ID system.

Further, for the effective use of IDS for networ surveillance, account must be taken of the poter tial impact of intrinsic factors gained from insigh in network operations on the performance of th IDS. The results also show that corporate ne works cannot rely entirely on currently availabl ID systems because of their inherent limitation The underlying factors responsible for this hav been articulated in this paper.

Finally, the deployment and monitoring tech niques that produced the best IDS performanc results in this study could serve as a useful guide i any IDS implementation program.

Appendix A. The RealSecure IDS software suit theoretical design performance specification

The RealSecure test suite provides for integrated network-based and host-based ID system available with over 450 built-in signatures. By design, it can monitor the IP traffic on the collision domain, or segment, the network engine resides. It can analyze 100% of IP traffic on 100 Mbps Ethernet segment with 60% sustained line utilization. It can process approximately 30,000 packets per second depending on engine configuration. If there are more packets per second being transmitted on the segment, RealSecure is still able to detect attacks but the reliability will decrease as the packet rate surpasses 30,000 pps. It is possible for RealSecure to process all packets on a segment that has more than 60% sustained utilization.

References

[1] B. Chapman, E. Zwicky, Building Internet Firewalls, O'Reilly and Associates, Inc., Sebastopol, CA, 1995.
[2] W.R. Cheswick, S.M. Bellovin, Firewalls and Internet security: repelling the wily hacker, Addison-Wesley, Reading, MA, 1994.
[3] D. Chapman, Network (in) security through IP packet filtering, in: Proceedings of the Third USENIX UNIX Security Symposium, Baltimore, MD, September, 1992.

[16] T.P. Martin, I.A. Macleod, J.I. Russell, K. Leese, B. Foster, A case study of caching strategies for a distributed full text retrieval system, Information Process Manager 26 (2) (1990) 227–247.
[17] P. Simpson, R. Alonso, Data caching in information retrieval systems, in: Proceedings of the 10th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New Orleans, LA, June 1987.
[18] A. Tomasic, H. Garcia-Molina, Caching and database scaling in distributed shared-nothing information retrieval systems, Tech. Republic, STAN-CS-92-1456, Stanford University, Stanford, CA, 1992.
[19] S. Gruber, J. Rexford, A. Basso, Protocol considerations for a prefi-caching proxy for multimedia streams, Technical document, AT&T Labs—Research, USA, September 1999.
[20] J.C. Mogul, Improving HTTP latency, in: Electronic Proceedings of the Second World Wide Web Conference: Mosaic and the Web, Chicago, IL, October 1994.
[21] J.C. Mogul, The case for persistent-connection HTTP, in: Proceedings of the ACM SIGCOMM'95 Conference on Communications, Architectures, and Protocols, Boston, August 1995.
[22] V.N. Padmanabhan, J.C. Mogul, using predictive prefetching to improve Worldwide Web latency, Computer Communication Review 26 (3) (1996).
[23] F.J. Burkowski, Retrieval performance of a distributed text database utilizing a parallel process document server, in: Proceedings of the International Symposium on Databases in Parallel and Distributed Systems, Dublin, Ireland, July 1990.
[24] T.R Couvreur, R.N. Benzel, S.F. Miller, D.N. Zeitler, D.L. Lee, M. Singhal, N. Shivaratri, W.Y.P. Wong, An analysis of performance and cost factors in searching large text databases using parallel search systems, Journal of American Society of Information Science 45 (7) (1994) 443–464.
[25] D. Hawking, Scalable text retrieval for large digital libraries, in: Proceedings of the First European Conference on Research and Advanced Technology for Digital Libraries, Pisa, Italy, September 1997.
[26] G. Trent, M. Sake, "Webstone: the first generation in HTTP server benchmarking, Technical report, MTS, Silicon Graphics Inc., February 1995.
[27] G. Banga, P. Druschel, Measuring the capacity of a Web server, Proceedings of USENIX Symposium on Internet Technology and Systems, California, USA, December 1997.
[28] J. Almeida, P. Cao, Measuring Proxy performance with the Wisconsin Proxy Benchmark, Technical document, Department of Computer Science, University of Wisconsin-Madison, USA, July 1997.
[29] IEEE 1355 DS link technology, available at: http://hsi.web.cern.ch/hsi/dshs/publications/rt97/html/rt97.html.

Charles Iheagwara is currently employed as an Information Technology Security Engineer at Edgar Online, Inc. in Rockville, Maryland, USA. He is licensed as a Professional Engineer (PE) and has many of the industry's top certifications including Cisco's CCNP and Microsoft's MCSE. He has completed the requirements for the Ph.D. degree in Computer Science at the School of Computing, University of Glamorgan, Wales, UK, where he specializes in Intrusion Detection Systems.

Andrew Blyth received his Ph.D. in 1995 from the University of Newcastle, UK. He is currently employed as a senior lecture at the School of Computing, University of Glamorgan, UK, where he specializes in Research in the areas of Information Warfare and Intrusion Detection Systems.

Appendix 5

"A Comparative Experimental Evaluation Study of Intrusion Detection System Performance in a Gigabit Environment." Journal of Computer Security, Vol 11 (2003) 1-33

Journal of Computer Security 11 (2003) 1–33
IOS Press

A comparative experimental evaluation study of intrusion detection system performance in a gigabit environment

Charles Iheagwara [a], Andrew Blyth [b] and Mukesh Singhal [c]
[a] *EDGAR ONLINE, INC. 8715 First Avenue, #1413D, Silver Spring, MD 20910, USA*
Tel.: +1 301 587 1408; Fax: +1 301 5871236; E-mail: ciheagwara@unateck.com
[b] *School of Computing, University of Glamorgan, Pontypridd, Wales CF37 1DL, UK*
Tel.: +44 1443 482245; Fax: +44 1443 482715; E-mail: ajcblyth@glam.ac.uk
[c] *Gartener Group, Department of Computer Science, University of Kentucky, 301 Rose Street, Lexington, KY 40506, USA*
Tel.: +1 859 257 3062; Fax: +1 859 323 3740; E-mail: singhal@cs.uky.edu

Intrusion detection systems' (IDS) effectiveness requires balancing characteristics and elements so they fit together in appropriate compromises to create good network security systems. One major gauge for IDS effectiveness is the ability to detect attacks within operational specifications. Gigabit IDS sensors as opposed to Megabit IDS sensors promise dramatic increase in component performance and functional opportunities, possibly leading to dramatically changed system balance and overall performance. The research described here examines the system benefits of using a single Gigabit IDS sensor instead of multiple Megabit sensors for a wide range of defined system attacks, network traffic characteristics, and for their contexts of operational concepts and deployment techniques. The experimental results are analyzed in the context of practical experiences in the operation of these IDS systems. The difference in architectural designs, deployment strategies and operational concepts that characterized their performance in exploiting the strengths of attack systems are discussed.

Keywords: Intrusion detection, network security

1. Introduction

The introduction of IDS security monitoring tools in recent years has come as a result of the inadequacies of traffic measurement tools to serve as effective security monitors. Traffic measurement tools do not usually offer support for security, nor do they allow active actions to be taken when an attack happens but they simply notify the administrators when an attack already took place. This is because measurement tools classify network traffic according to some specified static rules with defined thresholds. These thresholds are often not able to express complex traffic patterns or are not flexible enough to cover a whole subnet without having to define the same rule for all the hosts of the subnet. This results to a significant increase in the processing time of each received packet.

In recent years, in addition to intelligent filtering, there have been various developments in passive surveillance mechanisms to monitor network traffic for signs of

malicious or anomalous (e.g., potentially erroneous) activity. Such tools attempt to provide network administrators timely insight into noteworthy exceptional activity. Real-time monitoring promises an added dimension of control and insight into the flow of traffic between the internal network and its external environment. The insight gained through fielded network traffic monitors could also aid networks in enhancing the effectiveness of their firewall filtering rules.

Intrusion detection system is a security technology that attempts to identify and isolate 'intrusions' against computer systems. Different ID systems have differing classifications of 'intrusion'; a system attempting to detect attacks against web servers might consider only malicious HTTP requests, while a system intended to monitor dynamic routing protocols might only consider Routing Information Protocol (RIP) spoofing. Regardless, all ID systems share a general definition of 'intrusion' as an unauthorized usage or misuse of a computer system.

Typically, intrusions take advantage of system vulnerabilities [4] attributed to misconfigured systems, poorly engineered software, mismanaged systems, user neglect or to basic design flaw in for instance some Internet protocols. An intrusion detection system is a tool that attempts to perform intrusion detection. An intrusion detection system is a fast moving market with new players entering continuously. Commercial tools range from the widely available anti-viruses, to enterprise tools (e.g., Cisco/Netranger), to NT centric (e.g., Internet Security Services/RealSecure) and to configurable freeware (e.g., Network Flight Recorder). In fact such tools only detect suspicious events and report the intrusion and/or attempt to the operator. They do not include decision-making support for preventive or recovery actions.

Intrusion detection as an important component of a security system, complements other security technologies. By providing information to site administration, an ID system allows not only for the detection of attacks explicitly addressed by other security components (such as firewalls and service wrappers), but also attempts to provide notification of new attacks unforeseen by other components. Intrusion detection systems also provide forensic information that potentially allows organizations to discover the origins of an attack. In this manner, ID systems attempt to make attackers more accountable for their actions, and, to some extent, act as a deterrent to future attacks.

Effective implementation of IDS security facilities requires the ability of the IDS to integrate with existing network infrastructure and its interoperation with other security implementations on the protected network. At the same time, the requirements should not impose an usual burden on the IDS and thus impair its ability to be effective in capturing all traffic that originate from all specified network internally protected and Internet traffic or its compliance with specified security policy. In particular, the IDS should be able to carefully monitor those units that statistically originated most of the security attacks.

As with any other technology, there are pitfalls in the current implementation of commercially available IDS. The pitfalls include the issues of variant signatures, false positives and negatives alerts, data overload, difficulties to function effectively

in switched environments and scalability issues. The following is a brief description of the pitfalls.

Variants. While the ability to develop and use signatures to detect attacks is a useful and viable approach, there are shortfalls to only using this approach that should be addressed. Signatures are developed in response to new vulnerabilities or exploits that have been posted or released. Integral to the success of a signature, it must be unique enough to only alert on malicious traffic and rarely on valid network traffic. The difficulty here is that exploit code can often be easily changed. It is not uncommon for an exploit tool to be released and then have its defaults changed shortly thereafter by the hacker community.

Catch-up. New signatures can only be developed once an attack has been identified. Therefore between the creation of an attack and the deployment of a signature to detect the attack, a window of opportunity exists for an intruder to mount an attack with little to no chance of the attack being detected.

False positives. A common complaint is the amount of false positives an IDS generates. Developing unique signatures is a difficult task and often times the vendors will err on the side of alerting too often rather than not enough. This is analogous to the story of the boy who cried wolf. It is much more difficult to pick out a valid intrusion attempt if a signature also alerts regularly on valid network activity. A difficult problem that arises from this is how much can be filtered out without potentially missing an attack.

False negatives. Detecting attacks for which there are no known signatures. This leads to the other concept of false negatives where an IDS does not generate an alert when an intrusion is actually taking place. Simply put if a signature has not been written for a particular exploit, there is an extremely good chance that the IDS will not detect it.

Data overload. Another aspect, which does not relate directly to misuse detection but is extremely important is how much data can an analyst effectively and efficiently analyze. That being said the amount of data he/she needs to look at seems to be growing rapidly. Depending on the intrusion detection tools employed by a company and its size, there is a possibility for logs to reach millions of records per day.

Difficulties in switched environments. Network capture and analysis in a switched LAN environment usually means 'tapping' the switch's lines by using a 'mirror' port or deployment in other tapping configurations. In this approach, traffic is copied from one 'source' port to another destination or 'mirror' port.

It has been known that mirroring a full duplex source port may cause packet loss as traffic on the full duplex source port exceeds the available bandwidth of the mirror port.

Scalability Issues. In the last couple of years, there has been a significant increase in network traffic utilization. With this has come the introduction of Gigabit Ethernet technology to accommodate this increase in bandwidth – and thus the volume of traffic to be analyzed. The problem associated with this is that older IDS technologies that operate at 10 Mbps or 100 Mbps bandwidths are overwhelmed with the increase

in traffic volume. With Gigabit, the older IDS technologies become seriously overloaded. This problem is discussed in depth in Section 4.

This paper is intended to address one of the issues (scalability) mentioned above. Thus, in order to gauge the ability of currently available IDS to effectively scale to a very large size, the goal of the research in this paper is therefore:

1. To provide a probabilistic evaluation of the ability of Intrusion Detections Systems in a Gigabit environment to correctly identify attacks based on the signature analysis of the attacks;
2. To provide an evaluation of the performance of IDS in a Gigabit environment;
3. To analyze the impact of the characteristics associated with traffic flow on the performance of the IDS.

Prior research efforts in ID systems are discussed in the next section.

2. Related Work

Research into and development of automated Intrusion Detection Systems (IDS) has been under way for well over 12 years. By now a great number of systems have been deployed in the commercial or government arenas, but all are limited in what they can do. The creativity of attackers and the ever-changing nature of the overall threat to targeted systems have contributed to the difficulty in effectively identifying intrusions. While the complexities of host computers are already making intrusion detection a difficult task, the increasing prevalence of distributed networked-based systems and insecure networks such as the Internet has greatly increased the need for intrusion detection.

Previous and present IDS research that relate to the technological approach of Intrusion Detection Systems (IDS's) are identified into three categories:

1. Modeling – Misuse or anomaly detection;
2. Analysis; and
3. Deployment.

Detection is performed in the misuse detection model [1,3] by looking for specific patterns or sequences of events representing previous intrusions (i.e., looking for the 'signature' of the intrusion). It is a knowledge-based technique and only known intrusions can be detected. This is a more traditional ID technique, which is usually applied, for instance in the anti-virus tools.

In the anomaly detection model [3,12,13], detecting changes in the patterns of utilization or behavior of the system performs detection. Building a model that contains metrics derived from normal system operation and flagging as intrusive any observed metrics that have a significant statistical deviation from the model perform it. The approach is behavior-based and should be able detect previously unknown intrusions. It is in the research and development area in which currently innovative modeling

paradigms are explored which is inspired from biological systems. Pioneers in this area are from the University of New Mexico whose work is based on the idea that intrusion detection systems should be designed to function like the way the human natural immune systems distinguishes between 'self' from 'non-self' antibodies.

The main challenge with this approach, like for every behavior-based technique, is to model the 'normal' behavior of a process. Learning the activity of the process in a real environment can do this. Another approach, advocated by IBM research, consists of describing the sequences of audit events (patterns) generated by typical UNIX processes. Another method developed by Nokia is based on Kohonen Self Organizing Maps (SOM).

'Off-line' vs. 'Real-time' analysis [12] is another area where more conventional classification divides IDS's into systems which operate after the event and rely on analysis of logs and audit trails for preventive action and those that attempt real-time monitoring in the hope that precursor signs of abnormal activity give indication that corrective action is possible before real damage occurs.

Previous research on analytical methods for detecting anomalous or known intrusive activity [1,3,12,13] emphasizes on the different aspects of session activity within host boundaries given the fact that the primary input to intrusion-detection tools, audit data, is produced by mechanisms that tend to be locally administered within a single host or domain. However, as the importance of network security has grown, so has the need to expand intrusion-detection technology to address network infrastructure and services.

In the context of fault detection and diagnosis for computer network and telecommunication environments, network monitoring has been studied extensively by the network management and alarm correlation community [9,11,15,16]. The high-volume distributed event correlation technology promoted in some projects provides an excellent foundation for building truly scalable network-aware surveillance technology for misuse. However, these efforts focus primarily on the health and status (fault detection and/or diagnosis) or performance of the target network, and do not cover the detection of intentionally abusive traffic in distributed and switched environments. Indeed, some simplifications in the fault analysis and diagnosis community do not translate well to a malicious environment for detecting intrusions. For examples, assumption of stateless correlation, which precludes event ordering; simplistic time-out metrics for resetting the tracking of problems; ignoring individuals/sources responsible for exceptional activity.

As the scale of scientific research of IDS systems grows by leaps and bounds, so does the nature of IDS interoperation, architecture and implementation. Studies of ID systems attempting to address the issue of network surveillance include the Network Security Monitor developed at UC Davis [5], and the Network Anomaly Detection and Intrusion Reporter [10] developed at Los Alamos National Laboratory. Both perform broadcast LAN packet monitoring to analyze traffic patterns for known hostile or anomalous activity. Further, research by UC Davis in the Distributed Intrusion Detection System [24] and later Graph-based Intrusion Detection System [23] projects

attempted to extend intrusion-monitoring capabilities beyond LAN analysis, to provide multi-LAN and very large-scale network coverage.

Morris [17] and Maimon [14] investigated network traffic intensity measurement. Intensity measures distinguish whether a given volume of traffic appears consistent with historical observations. These measures reflect the intensity of the event stream (number of events per unit time) over time intervals that are tunable. Alternatively, a sharp increase in events viewed across longer durations may provide insight into a consistent effort to limit or prevent successful traffic flow. Morris investigated intensity measures of transport-layer connection requests, such as a volume analysis of SYN-RST messages, which could indicate the occurrence of a SYN-attack [17] against port availability (or possibly for port scanning). Maimon explored intensity measures of TCP/FIN messages as a variant [14] considered to be a more stealthy form of port scanning.

In their studies [14,17], the authors contend that monitoring overall traffic volume and bursty events by using both intensity and continuous measures provides some interesting advantages over other monitoring approaches, such as user-definable heuristic rules that specify fixed thresholds. In particular, the intensity of events over duration is relative in the sense that the term 'high volume' may reasonably be considered different at midnight than at 11:00 a.m. The notion of high bursts of events might similarly be unique to the role of the target system in the intranet (e.g., Web server host versus a user workstation).

Traffic Analysis with Signature Analysis has been studied [12,18,19,21,26]. Signature analysis is a process whereby an event stream is mapped against abstract representations of event sequences known to indicate the target activity of interest. Determining whether a given event sequence is indicative of an attack may be a function of the preconditions under which the event sequence is performed.

The use of coding schemes for representing operating system penetrations through audit trail analysis was also the focus of other research works [12,18,19]. Using basic signature-analysis concepts, it was shown that some detection methods could support a variety of analyses involving packet and transport datagrams as event streams. For example, address spoofing, tunneling, source routing [21], SATAN [26] attack detection, and abuse of ICMP messages (Redirect and Destination Unreachable messages) could all be encoded and detected by signature engines that guard network gateways.

The advent of large scale commercial intrusion detection systems tend to have given a relative assurance to the information technology community that has been very anxious to maximize the use of these highly advertised ID systems as added armor to secure network systems. Many IDS products have been deployed in commercial and corporate networks. With this has come a shift in research focus in so many areas. One such area is the IDS performance.

IDS evaluation studies [8,22] treat the relationship between deployment techniques and attack system variables and the performance of the IDS.

Richards [22] evaluated the functional and performance capabilities of the industries' leading commercial type IDS. In the areas tested, the performance of the IDS

was rated based on their distinctive features, which were characterized into different performance indexes. The research work represented a new direction for ID systems in that it moved the focus away from scientific concepts research to performance evaluation of the industries' best products. However, the study was limited to a small proto design isolated and non-switched network which did not reveal the impact of packet switching on the accuracy and ability to capture attack packets in their entirety. Iheagwara and Blyth [7] expanded this effort to an evaluation study of the effect of deployment techniques on IDS performance in switched and distributed system. They demonstrated that monitoring techniques could play an important role in determining the effectiveness of the IDS in a switched and distributed network.

Porras and Valdes [20] discussed IDS failures in terms of deficiencies in accuracy and completeness, where accuracy reflects the number of false positives and completeness reflects the number of false negatives.

All of the above research works predated the advent of Gigabit network and the scalability issue associated with IDS deployment in Gigabit environment thus opening up a new area of research focus.

The problem here is that with the advent of Gigabit Ethernet not only is there a significant increase in bandwidth – and thus a significant increase in the volume of traffic to be analyzed – but also a move into the realms of the purely switched networks. Because in the promiscuous mode sensors can only see traffic on its own segment, and in a switched environment, every connection to the switch is effectively, a single segment. In the older technologies of 10 mbps or 100 mbps bandwidths, this can be overcome by the use of network taps or mirroring all the switch traffic to a span port, to which the IDS sensor is attached. But with Gigabit networks, the result would be a seriously overloaded sensor. Currently suggested solutions include building an IDS technology into the switch hardware itself that will allow the sensor to grab traffic directly from the backplane or in the alternative move to a pure Network Node IDS implementation where the agents are concerned only with the traffic directed at the host on which they are installed.

The currently available commercial IDS were designed to accommodate traffic with bandwidth not exceeding 100 Mbps. Deployment of these IDS on Gigabit traffic presents scalability problem and has not been independently evaluated (at least not in any published scientific literature).

In Section 3, we describe the research work conducted to evaluate and compare the performance of multiple 100 Mbps IDS sensors and a single Gigabit sensor in a gigabit network environment.

3. Experimental Work

Objectives

The goal of our experiment is two fold:

1. To provide an evaluation of the performance of IDS in a Gigabit environment; and
2. To analyze the impact of the characteristics associated with the traffic flow on the performance of the IDS.

3.1. Evaluation criteria

The basic performance indicators of the IDS are reflected in the success or failure of event analysis, which are quantitatively measured for qualities such as accuracy and performance: both are assessable through testing. A more difficult but equally important metric to assess is completeness. In this case, inaccuracy is reflected in the number of legitimate transactions flagged as abnormal or malicious (false positives), incompleteness is reflected in the number of harmful transactions that escape detection (false negatives), and performance is measured by the rate at which transactions can be processed.

Within the limits of our experiment, the evaluation criteria used is the percentage of attacks captured by the IDS against the tunable experimental parameters, i.e., bandwidth, throughput, traffic characteristics and attack signatures.

3.2. Framework

The study is based on the premise that the only true way to scale an IDS effectively is to use a flow based switch to 'split' or 'load balance' the sessions or connections etc. across multiple IDS's.

Using this concept, we tested the IDS against a Gigabit traffic background by spanning all traffic (attack, victim, background) to a gigabit port that feeds the traffic to a switch. The switch then feeds the traffic to one attached gigabit IDS sensor or load balances it to multiple 100 Mbps IDS sensors. The derivative benefits here are:

1. The span traffic can be incremented up to or less than 100 mbps and the signature libraries tested against multiple 100 Mbps sensors versus one gigabit sensor; and
2. The span traffic can be incremented up to one gigabit and the IDS performance is tested with gigabit sensor versus multiple 100 mbps sensors.

3.3. Experimental method

The experimental model envisages mimicking a series of defined and designated attack sets that are incorporated into the software suite of the ID system. The attack set classification is given Section 3.9. For each attack, we used a specific attack signature exploit in the areas tested. The attack list is not meant to function as a complete list of attacks, but rather as a limited test suite that has been designed to test the performance capabilities of IDS systems on a Gigabit network.

In each test, we injected specifically configured attack packets onto a test network, on which the subject ID system was running in order to establish interactions between injected packets and the host target of attack. To create the needed TCP connections for the subject IDS to monitor, the target host was the explicit addressee of all the attack packets. In addition, the target host also acted as a control for the experiments. The target's response to injected attack packets allowed us to empirically record the behavior of the performance of the ID system in all test categories.

We observed the attack detection performance and functional characteristics of the ID system and the system's underlying TCP/IP implementation by tracking the subject's monitoring console output and considering the specific types of packets used for the test.

In order to provide experimental Quality of Service (QoS), the following, which could impact the test results, were considered:

(i) Environmental test condition;
(ii) Flow control; and
(iii) Switch culture.

3.3.1. Environmental test condition

The environment created for the testing was designed to mimic a real network that simulated high bandwidth and realistic traffic with a few test systems. Background traffic with modest distribution of packet size to avoid any testing bias was created with valid headers and checksums so that the switches would never pass them to the intrusion detection system (IDS) sensors. Spanning tree was disabled on the switch connecting the network sensors.

3.3.2. Flow control

Flow control was established by implementing full client/server flows for background traffic by using UDP requests and replies transmitted between the ML7710 SmartBits cards (see the Appendix for a description). This was important to prevent flooding large quantities of test traffic to unknown destinations. Full duplex network analyzers were used to validate and troubleshoot the test environment (from bad patch cables to flooding traffic on switch ports) in order to remove any restriction on traffic flow that may impair the functionality of the IDS or AppSwitch (see the Appendix for a description).

3.3.3. Switch culture

The switch was cultured to treat the simulated systems as real by issuing layer 3 ARP packets from the SmartBits cards before every round of testing. The attack traffic was live with real source and destination MAC and IP addresses.

3.4. Test bed

The test bed is represented in two logical diagram Figs 1 and 2. The two figures are essentially the same except for the placement of the sensors. In the test diagram

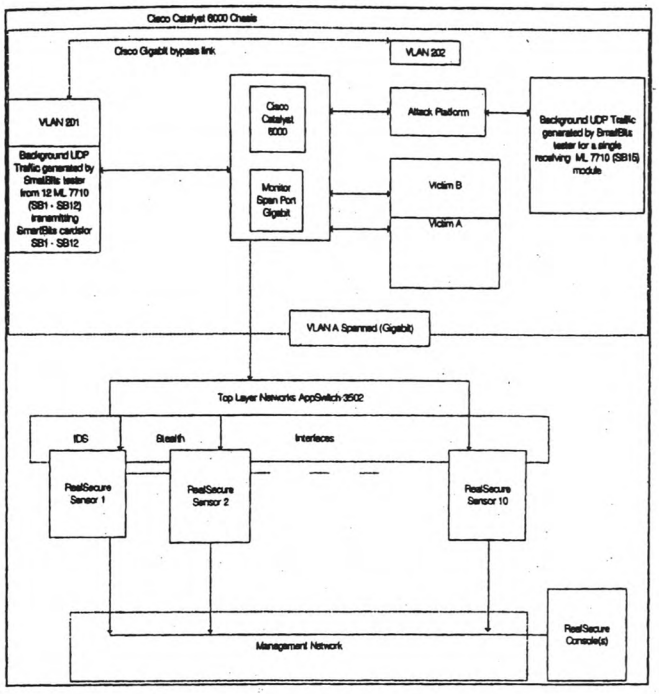


Fig. 1. RealSecure IDS test environment logical diagram.

(Fig. 1), ten Realsecure sensors were used to monitor the traffic stream coming from the AppSwitch 3502 that they are connected to. Whereas in Fig. 2, only one NetworkIce Gigabit sensor is used. The rest of the test setup is identical.

In the logical test diagrams, 13 SmartBits network cards from the SmartBits tester background traffic generator were connected to 13 (100 Mbps) ports on a Cisco Catalyst 6000 48 port module. Traffic flow between the SmartBits tester, Cisco catalyst 6000, attack and target systems depicted in Figs 1 and 2 were organized into two virtual local area networks (VLANS) 201 and 202. VLAN 201 consists of UDP background traffic from the SmartBit tester and traffic from the 13–100 Mbps and 4 Gigabit ports on the Cisco catalyst 6000. VLAN 202 consists of traffic from the attacker and target systems. Traffic from VLAN 202 was inserted into VLAN 201 via a Gigabit crossover cable to disburse the attack throughput the background traffic before forwarding it to the AppSwitch. Addition of the Gigabit ports to VLAN 201 ensured enough bandwidth within the VLAN.

Systems description of the set up for the attacker and victims are given in the Appendix. The Cisco catalyst 6000 48 port module consists of 44 100 Mbps and 4 Gigabit ports.

All traffic from VLAN 201, which included attack traffic, was mirrored to a Catalyst Gigabit SPAN port. Traffic from the Catalyst Gigabit SPAN port fed the App-

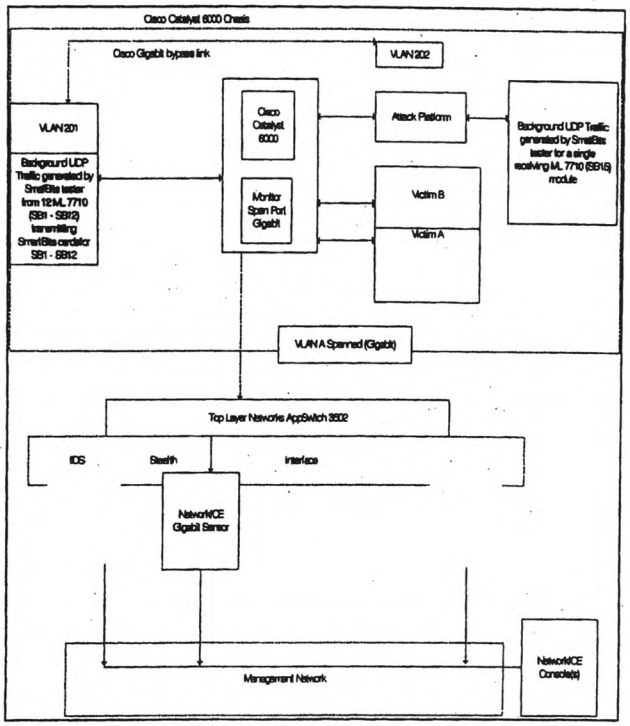


Fig. 2. NetworkIce IDS test environment logical diagram.

Switch Gigabit input which then flow mirrored the traffic to the RealSecure network sensors connected to a maximum of 10, 100 Mbps ports or the NetworkICE gigabit sensor. The IDS sensors monitored that traffic via their stealth interfaces and reported attacks to the management console via their second interface.

3.5. Test procedure

In the performance tests, we used the AppSwitch and Cisco switches, which only pass valid traffic to real systems with both the attack and the background traffics containing real source and destination MAC addresses.

Once the test bed and environmental test control was set up, control runs were completed against each component to ensure that they are functioning properly. In order to validate the integrity of the test, the following control methods were employed:

- Monitoring of the test network with background traffic or attacks to ensure there is no erroneous traffic other than bridging protocols.

- Generation of each level of background traffic, from 100 Mbps to 1 Gbps and verified the traffic generation settings by monitoring with a network LAN analyzer.
- With no background traffic, attacks used against the RealSecure sensors were generated to be sure their policies are properly configured to alert on the attacks and that they are communicating with their consoles.
- For ease of execution, we used either Network Associates, Inc. Cybercop Scanner IDS attack testing suite or created an interactive TCL and Casl scripts to log into the appropriate service to run each attack. There were 10 of each attack type – HTTP, FTP, and SMTP launched against each of the two target systems during each run for a total of 30 attacks per target and 60 attacks for the entire test.
- The RealSecure kill response was not used as the Cisco span ports were configured only for uni-directional traffic in RX only mode.

The tests were preceded by a series of baseline tests against the target host to ensure that the subject IDS was configured and functioning properly according to design specifications. In almost all test cases, a process on the target host ran which accepted incoming TCP connections on the HTTP port and printed any input obained from the machine's TCP stack. By examining the output of this process, we were able to deduce whether the subject IDS should have detected the attack based on the network conditions we created.

The sequence of the test procedure is as follows:

1. Generation of ARP packets with SmartBits to update switch ARP cache.
2. Generation of 100 Mbps (10% of Gigabit) background traffic and verifying with a network analyzer.
3. Generation of attacks against both target systems.
4. Recording how many of the total attacks were caught by the RealSecure sensor.
5. Discontinuation of traffic generation.
6. Clearing of the RealSecure console display.
7. Repetition of steps 3–6 two more times for a total of three trials at each utilization.
8. Repetition of steps 1–7 with a single sensor for background utilizations of 200 Mbps (20% of Gigabit), 300 Mbps (30% of Gigabit), 700 Mbps (70% of Gigabit), and 800 Mbps (80%).
9. Addition of 1 sensor to the AppSwitch, for a total of 2 and repeating steps 1–8.
10. Addition of one sensor at a time until the IDS sensor is able to capture all attacks at the highest utilization.

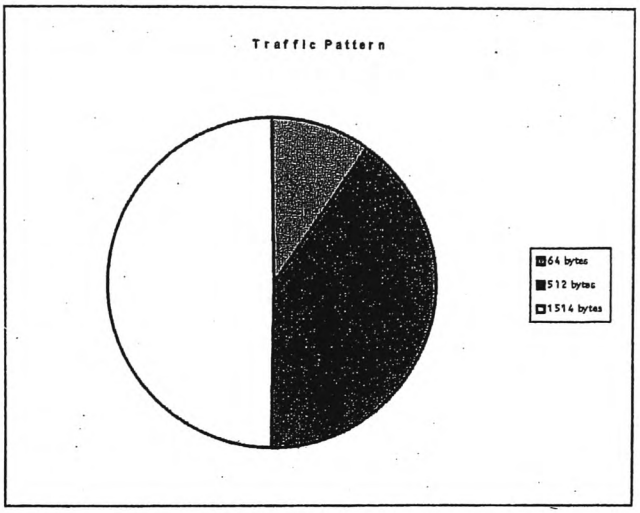


Fig. 3. Traffic distribution profile.

3.6. *Traffic generation*

To generate realistic background traffic during the testing, the SmartBits tester was configured to generate 3 UDP sessions from each of the 12 transmitting ML7710 modules for a total of 36 sessions or streams. A single receiving ML7710 module was configured to transmit infrequent replies to the transmitting modules to maintain the MAC and ARP forwarding databases in the AppSwitch and Catalyst and thus avoid flooding of traffic (a session is defined as a flow of UDP traffic). The SmartBits cannot currently generate real TCP sessions as of this writing with a unique source or destination MAC address, source or destination IP address, and source or destination UDP port number. Stated differently, any deviation in any of the above field values constitutes a unique session. A variety of UDP port numbers were used to make the traffic more realistic.

A component level description of the test bed is given below and the specification of the AppSwitch is given in the Appendix.

3.7. *Traffic characteristics*

The profile of background traffic consisted of three packet sizes: 64 bytes, 512 bytes and 1514 bytes. Of the 12 transmitting ML7710 SmartBit cards, 4 sent 64 byte packets, 4 sent 512 byte packets, and 4 sent 1514 byte packets. The total offered load for each test sequence is defined in Mbps-megabits per second. The total offered load was controlled by manipulating the frames per second transmit parameter on each of the transmitting ML7710 SmartCards. Utilization for each test run was verified with multiple network analyzers.

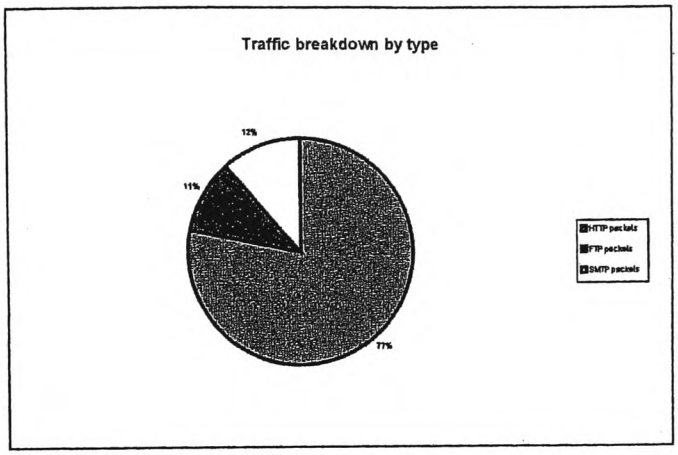


Fig. 4. Traffic distribution by type.

Packet size traffic distribution was as follows: 50% of the total load in Mbps from the 1514 byte packets, 40% of the total load from 512 byte packets, and 10% of the total load from the 64 byte packets. This profile was derived from on-going studies [2] that measure typical Internet traffic patterns at two MCI backbone points in the USA performed by CAIDA – Cooperative Association for Internet Data Analysis.

3.8. Attack signatures

We used the standard 5.0 attack detector policy. Each signature was set to count the number of packets it triggers. All signatures for all attacks were enabled with alert console. The log to database responses was also enabled. Other attack signatures not needed in the attack (ARP, IP Duplicate, IP Frag, IP ProtocolViolation, IP Unknown Protocol, SourceRoute and TCP_Overlap_Data) were disabled because some SmartBits packets are recognized as non-standard by the RealSecure network sensor.

Some signatures (port scans and indexed attack signatures) were not tested because load balancing is known to diminish their effectiveness across multiple sensors.

3.9. Classification of test attack set

The attacks used were a mixture of Web, email and ftp attacks representing a mixture of typical, multi-packet attacks using a variety of protocols that are common in most networks today [27]. The purpose of the attack set is to function as a test suite. The common vulnerability exploits cross-reference is shown in Table 1. The ratio of packet generation of each attack type shown in Fig. 4 is:

Table 1
CVE cross-references

Name of Vulnerability	CVE/CAN Reference
HTTP WWW-COUNT-CGI-BIN	CVE-1999-0021
HTTP-PHF	CVE-1999-0067
HTTP-DOTDOT	CAN-1999-0776
HTTP-IE3URL	CVE-1999-0280
HTTP-APACHE-DOS	CAN-1999-0107
SMTP_PIPE	CAN-1999-0163
SMTP_VRFYBO	CAN-1999-0531
SMTP_EXPNBO	CAN-1999-0531
SMTP_DEBUG	CVE-1999-0095
FTP_ARGS	CAN-1999-0076
FTP_ROOT	CAN-1999-0527

- HTTP packets – 77%;
- FTP packets – 10.5%;
- SMTP packets – 11.5%.

The attack sets are:

- FTP
  - ftp_args,
  - ftp_root
- HTTP
  - http_phf,
  - http_ie3url,
  - http_dotdot,
  - http_coldfusion,
  - http www-count cgi-bin,
  - http_apache_dos
- SMTP
  - smtp_pipe,
  - smtp_vrfybo,
  - smtp_expnbo,
  - smtp_debug.

3.10. Experimental results

The results obtained in the tests are presented in Tables 2 and 3. The main objective of using intrusion detection for real-time monitoring of TCP/IP-based networks traf-

Table 2
IDS test results with 'NetworkICE' sensor

% Utilization, Mbps	% Detection
100	91
200	89
300	86
400	76
500	74
600	71
700	53
800	41
900	N/A
1000	N/A

Table 3
IDS test results with multiple IDS sensors

Utilization (GBps)	% Detection 1 IDS	% Detection 2 IDS	% Detection 3 IDS	% Detection 4 IDS	% Detection 5 IDS	% Detection 6 IDS	% Detection 7 IDS	% Detection 8 IDS	% Detection 9 IDS
0.1	76	83	87	88	91	92	94	94	96
0.2	44	69	77	80	88	89	90	91	93
0.3	32	49	73	83	87	88	88	89	90
0.4	21	36	60	74	83	80	87	87	88
0.5	14	28	44	71	81	79	82	84	86
0.6	12	21	42	62	67	79	82	83	83
0.7	10	19	32	58	63	75	75	75	76
0.8	10	14	29	40	62	74	74	74	74
0.9	10	10	10	10	10	10	10	10	10
1	NA	NA	NA	NA	NA	NA	NA	NA	NA

fic is the detection, quantification and analysis of malicious (or exceptional) network traffic. This is accomplished by examining captured packets, which individually represent parsable activity records, where key data within the header and data segment can be analyzed and/or heuristically parsed for response-worthy activity. In line, the test results represented in Figs 5 and 6 show the number of attacks detected. We have used the number of attacks detected instead of the number of packets captured since the ultimate in any instance or scenario attack is quantification of attack detection.

When interpreting the test results of two IDS products with different architectures, a standard yardstick for their comparison must be well defined. Here a common metric of comparison is to use the nominal attack detection capability specified by the product designers. The RealSecure sensor is designed to operate at 100 Mbps

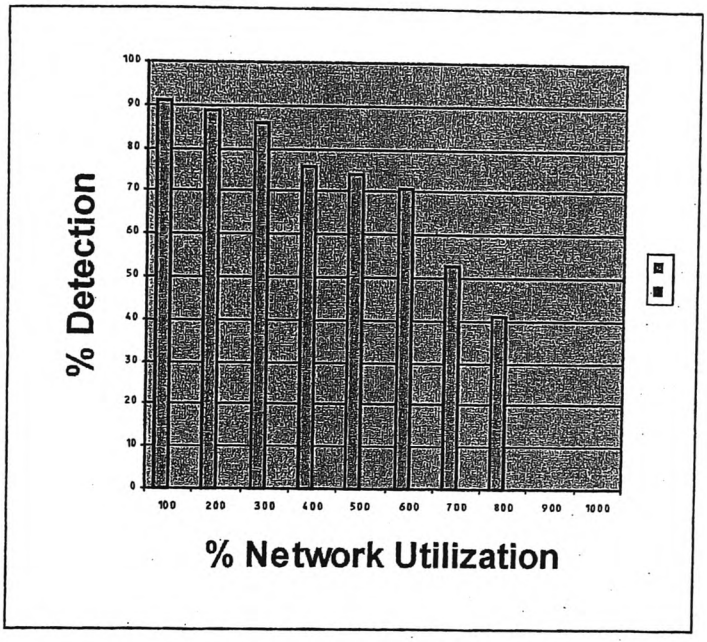


Fig. 5. Probability of detection vs. % utilization with NetworkICE gigabit sensor.

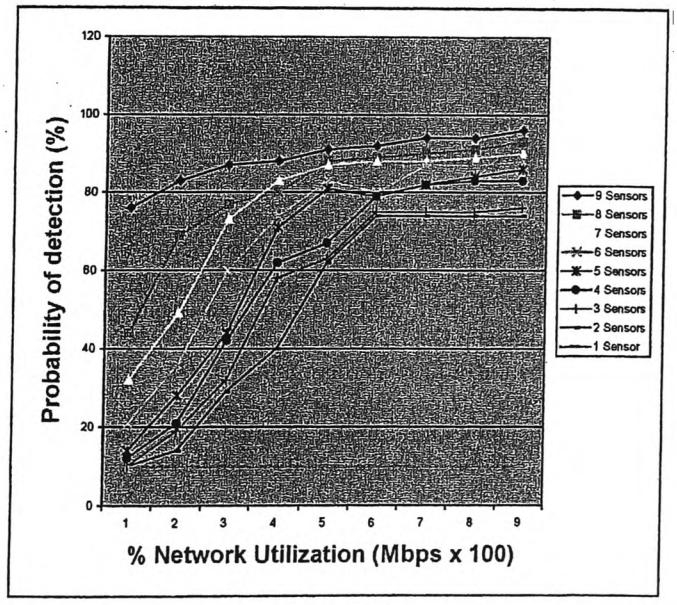


Fig. 6. Probability of detection vs. percent utilization with RealSecure Multiple sensors.

Table 4
IDS sensor vs % detection

Mbps	% Detection	# Of Net -workIce sensor	# Of RealSecure sensor
100	91	1	5
200	89	1	6
300	86	1	5
400	76	1	4
500	74	1	4
600	71	1	5
700	53	1	4
800	41	1	4

although best performance in actual networks is in the range of 30 to 40 Mbps. The NetworkIce Gigabit sensor is designed to operate at 100% detection rate in Gigabit traffic, which technically is from 300 Mbps to 1 Gbps. This essentially means that 3 RealSecure sensors should match the detection rate of 1 NetworkIce sensor at 300 Mbps or at 800 Mbps, which is more realistic than 1 Gbps, 8 Realsecure sensors will match the performance of the NetworkIce sensor.

We can use the above standards to discuss the data in Table 4. Essentially, the data clearly shows that from low to middle level Gigabit traffic (100–600 Mbps), stream between 4 to 6 RealSecure sensors or 1 NetworkIce sensor can detect more than 75% of the attacks. Above this point, detection rate drops to undesirable levels although only about 4 RealSecure sensors will be needed to match the detection rate of 1 NetworkIce Gigabit sensor.

Individually, the percentage of detection shown in Figs 5 and 6 demonstrate that the NetworkIce sensor was more effective in detecting attacks from 100 Mbps to 300 Mbps while the Realsecure sensor was more effective at network loads above 300 Mbps. This is due to the fact that the number of the RealSecure sensors matching the detection rate of the NetworkIce was smaller than expected by the design metrics.

Generally, as is shown in Fig. 7 the detection rate for both types of IDS decreases with increase in network utilization. The decrease is more pronounced above 600 Mbps of network utilization.

An accurate determination of false alarms could be a daunting task as false alarms could be overwhelming. In the experiments, closer analysis of the test data of all attacks showed that the number of false alerts generated is in the order of 17%. This figure was obtained from the recording and analyses of the log to database responses, which was enabled as described in Section 3.8.

4. Analysis

In analyzing the results, there are many factors that impact the performance of the IDS sensor that needs to be considered. Some of which are due to differences

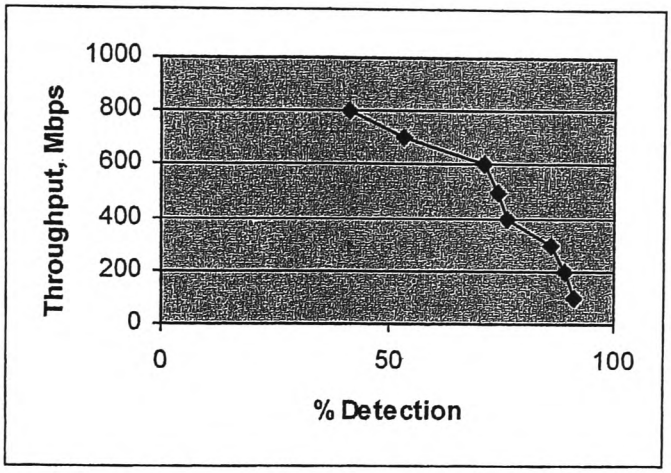


Fig. 7. Throughput vs % detection.

in the IDS's design while others are due to a variety of factors. The analysis must also take account of the experiences in the operation of the IDS products in real world environments. Experience has shown that the IDS performance and its stability (i.e., ability to function within design limits without failure) are determined by the following:

- Design limitations
- Traffic rate (number of packets per second)
- Traffic type (i.e., HTTP, FTP, SMB, SMTP, etc.)
- Packet size
- Session lengths
- % of fragmentation
- Number of sessions/Hosts
- Number of signatures active
- Workstation hardware
- Half/Full duplex transfer mode

4.1. Design limitations

Evaluation of test results requires a methodical analysis of the many factors that could affect the IDS performance in actual network environments. This is because it is possible for the IDS to perform differently even under the same parametric specifications but different environmental contexts.

For instance, there could be cases of attaining a 100% detection rate when 100% of the traffic was scripted, but when background/normal traffic or encrypted traffic is used or added the performance goes down. Equally, it is possible to toss 40 Mbps

of traffic at the IDS that won't phase it, and another 40 Mbps that will phase it. In this regard, experience has been that what breaks an IDS is more often packets per second than Layer 7 content [8], although both are relevant.

Generally, there are three bottlenecks that affect the performance of the IDS in real world environments.

- Raw sniffing speed
- Signature degradation
- Memory usage.

4.1.1. Raw sniffing speed

Sniffing speed as a measure of how much packets per seconds can be captured is a very important factor when evaluating the performance of ID systems. This is due to the fact that this could be used as a baseline when determining the maximum packet capture/second in order to quantify the operational bandwidth limits after which the performance of the IDS begins to diminish. Thus, it is a valuable measure that shows the maximum load at which the IDS will still operate effectively. The figures available from some IDS vendors as performance bottlenecks are:

- 200 000 packets/second for Cisco's Secure;
- 70 000 packets/second for Intrusion.com's Gigabit sensor; and
- 700 000 packets/second for ISS's NetworkICE Gigabit sensor.

Of interest here is NetworkIce's 700 000 packets/second sniffing rate. This means that given optimum conditions, the Gigabit sensor's engine should be able to process 700 000 packets per second. The RealSecure sensor will not sniff beyond 100 000 packets/second. It is assumed that the packets related to the above numbers are true for all (typical) packet sizes.

Consequently, what this means is that seven RealSecure sensors will be required to match the performance of NetworkICE's sensor for a 700 000 packets/second capture in any given identical context.

In analyzing the results, we used the vendor provided data as the baseline reference in setting a comparison standard. Evaluation of the IDS products was based on the percentage of detection of attacks.

It should be noted that this could take a different meaning if we factor in packet size in the packets per second discussion. For instance, it is possible to address 1-Gbps networks by pushing through 1500-byte packets at 70 000 packets/second. This means using the smaller packet sizes that are likely to be seen in the real world means that IDS product is unlikely to exceed 200-mbps.

4.1.2. Signature degradation

The second bottleneck is that Network Intrusion Detection System (NIDS) analysis at high rates comes with signature degradation. Most NIDS use 'pattern-matching routine' (signature-based), which slows/degrades with successive addition of signatures. Network ICE uses 'state-based protocol-analysis', which means that it does

not slow down as you add signatures because it follows a decision tree. This means that when running in the 1-Gbps ranges, all signatures can be enabled. To solve the problem of false positive alerts, filters can be set up on some signatures, thereby making it not necessary to remove signatures in order to performance tune.

The RealSecure IDS uses the pattern matching technique that somewhat impairs its functionality because the pattern matching technique degrades with an increase in the number of active signatures.

The theory behind interpreting IDS performance, by comparing 'state-based protocol-analysis' vs. 'pattern-matching' techniques could be explained from the perspective of the two fundamental advantages that state-based protocol-analysis has over pattern-matching in regards to performance:

1. More efficient processing of traffic.
2. Scales better as you add more signatures.

A good example would be to compare how an IDS looks for RPC exploits. A pattern-matching system looks for patterns on ranges of ports where RPC programs typically run. For example, it might look on ports in the range 634 through 1400 for the AMD exploit. In contrast, a state-based system can remember which ports the AMD service is running on, and only test the AMD signatures on those ports that are actually running AMD. If no system on the network is running AMD, then a state-based system will never test network traffic for those signatures.

The theory behind this is that a pattern-matching system doesn't know the contents of the packets, and must match that packet for many different patterns. In contrast, a protocol-analysis system knows the contents of the packet, and only tests signatures that apply to those contents.

Given an average packet, a pattern-matching system might have to match for 10 different patterns within that packet. In contrast, on average, a state-based protocol-analysis system tests less than 0.1 signatures per packet.

This doesn't come for free: the state-based protocol-analysis that knows whether or not it should test for signatures itself costs the same as testing for a couple of signatures. Thus, the per-packet cost for pattern-matching might be 10 signatures, and the per-packet cost for state-based protocol analysis might be 2 signatures.

The second part of the theory is that for pattern-matching systems, the more signatures you add to the system, the slower the system becomes. If you look in the documentation for the average sensor, it will have a comprehensive discussion on how to remove signatures in order to improve performance. This isn't applicable to a state-based protocol-analysis system.

A good example is to consider looking for Telnet login strings. There are many well-known login names that rootkits will leave behind on the system. A pattern-matching system must scan all Telnet traffic for all these patterns – the more patterns you add, the slower it becomes.

In contrast, a protocol-analysis system will decode Telnet and extract the login name. It can then lookup the name in a binary-matching tree or a hash table. The

difference is that a pattern-matching system must match for patterns within network traffic, which scales poorly. In contrast, a protocol-analysis system pulls out a field from network traffic, and matches that field within an internal table, which scales very well to log in name.

Again, not in the Telnet example that a username signature is only tested against the username field – another demonstration of the first point that a packet is only tested for a signature when needed, and not when it isn't needed.

This is the theory behind the comparison. In practice, there are a lot of issues that can become more important. For example, CPU speeds are doubling every year.

The limitations imposed by signature issues are discussed in Section 3.8 and only apply to the scope. However, there is no known impact of signature degradation on the performance of the RealSecure sensors because we did not run into signature overloads.

4.1.3. Memory usage

All currently available network intrusion detection system (NIDS) track TCP connections because they have to reassemble them, or risk being evaded. The problem here is that Gigabit networks in most cases have millions of outstanding TCP connections. This causes most boxes to fail over. The architecture of the NetworkICE sensor incorporates memory-saving techniques that optimize memory consumption in preference to speed. So also does the ReakSecure architecture hold well with memory consumption.

Therefore, within the context of our test studies, memory usage was not a problem obviously due to the fact that the architecture of both IDS systems does well with memory usages.

From the above discussions, it is clear that the design-related performance bottlenecks did not impair the performance of the IDS products evaluated due to the scope of the experiments. That being said, in real network environments, these could impact the IDS performance especially from signature overload of the RealSecure IDS sensors.

4.2. Typical traffic

When evaluating the performance of the IDS, network throughput is important. This is commonly expressed in either Megabits (Megabytes) or Gigabits (Gigabytes). A crucial question is how many megabits (Mb) can the IDS handle before its performance nosedives?

Gauging the performance of the IDS is a function of many variables. For instance, if a packet of 1500 byte that is invalid or contains no interesting information is loaded on the network at a high rate, it will not be effective in testing the IDS. To characterize the true bandwidth limits within which the IDS is effective, the processing power of the IDS must be tested using properly configured packets. It is not just enough to send 100 Mbits of 512 byte packets with a traffic generator. There is the need for a

traffic that is close or identical to real traffic from real machines that is repeatable; yet still random enough that one does not end up with the vendors catering to bandwidth benchmark. That is why it is necessary to use traffic that is identical to real traffic from real machines in a performance evaluation. We applied this principle to our tests by mimicking a 'typical traffic' as discussed in Section 3.7.

Another dimension here is the variable nature of traffics on most networks. Traffic varies greatly from network to network. Internal enterprise networks might see a lot of SMB, NFS, SNA, and SQL network traffic. For example, while external/DMZ networks might see mostly HTTP, FTP, SMTP, and the occasional SSH session, a university network will see a lot of HTTP, FTP, SSH, SMTP, IMAP, POP, Napster, IRC, and a myriad of other protocols that you won't see in the average corporate space and a carrier network will see everything from HTTP traffic to BGP updates, and every other protocol that goes across the network.

The point is that there isn't really an easy way to say 'typical traffic'. One might be able to craft some baseline assumptions on what university traffic looks like, what internal corporate traffic looks like, what DMZ/external corporate traffic looks like, what ISP traffic might look like, etc., but environments are so wide and varied that there is no 'one size fits all' approach to traffic modeling. For example, sending 100 Mbits of a typically used protocol (like HTTP) could crush an IDS that wouldn't produce the same result with for instance, 500 Mbits of UDP traffic on a non-standard port.

To ensure that the test traffic fits into the 'typical traffic' type, we used a representative mix of traffic as discussed in Sections 3.7 and 3.8. This shows that the test results would not have been different in a real word context bearing other factors.

4.3. Packet size

Instances exist when the attainment of maximum (%100) utilization will have different meaning depending on the context. For instance, in analyzing an output such as the one depicted in Table 5 (chart) [25], 100% utilization could be 64 byte frames at 14 880 pps, or 1518 byte frames at 812 pps. There is a big difference here because processing-wise, the two are not equal. We can not relate the above to capacity utilization, because we have less than 50% of the information required to simply say that utilization is 'X' Mbps.

Table 5

Data Field Size	Max Frames/sec	Max Data Field Bits/sec
46 (64)	14 880	5 475 840
64 (82)	12 254	6 274 084
128 (146)	7530	7 710 720
256 (274)	4241	8 706 048
512 (530)	2272	9 306 112
1024 (1042)	1177	9 641 984
1500 (1518)	812	9 744 000

In the tests, we used varying packet sizes as discussed in Section 3.7. This also indicates that bearing other factors, the results will hold in real world traffic environments.

4.4. Number of sessions

The complexity of analyzing IDS performance increases with another variable – number of sessions. This is because many ID systems have to track state and to a certain extent; the number of sessions is a huge factor. In this regard, 4880 pps between two hosts is very different from 14 880 pps between 5000 hosts. This is demonstrated by the fact that there have been instances when ID systems starts degrading in performance at 6500 pps under 35 Mbps network load with little chance of recovering based on the number of sessions observed by the IDS.

4.5. Other factors

There are other factors that could affect the performance of the IDS that did not affect our test results due to the size, scope, nature, and environmental test conditions used in our tests.

Network ID systems work by predicting the behavior of networked machines based on the packets they exchange. The problem with this is that a network monitor that is not active cannot accurately predict whether a given machine on the network is even going to see a packet, let alone process it in the expected manner. The existence of a number of factors could make the actual meaning of a packet captured by IDS ambiguous. These can be considered as follows:

1. A network IDS is typically on an entirely different machine from the systems it's watching. Often, the IDS are at a completely different point on the network. The basic problem facing a network IDS is that these differences cause inconsistencies between the ID system and the machines it watches. Some of these discrepancies are the results of basic physical differences, others stem from different network driver implementations. For example, consider an IDS and an end-system located at different places on a network. The two systems will receive any given packet at different points in time. This difference in time is important; during the lag, something can happen on the end-system that might prevent it from accepting the packet. The IDS, however, has already processed the packet thinking that it will be dealt with normally at the end-system.
2. IP packet with a bad UDP checksum will not be accepted by most operating systems. Some older systems might. The IDS needs to know whether every system it watches will accept such a packet, or it can end up with an inaccurate reconstruction of what happened on those machines. Some operating systems might accept a packet that is obviously bad. A poor implementation might, for example, allow an IP packet to have an incorrect checksum. If the IDS don't know this, it will discard packets that the end system accepts, again reducing the accuracy of the system.

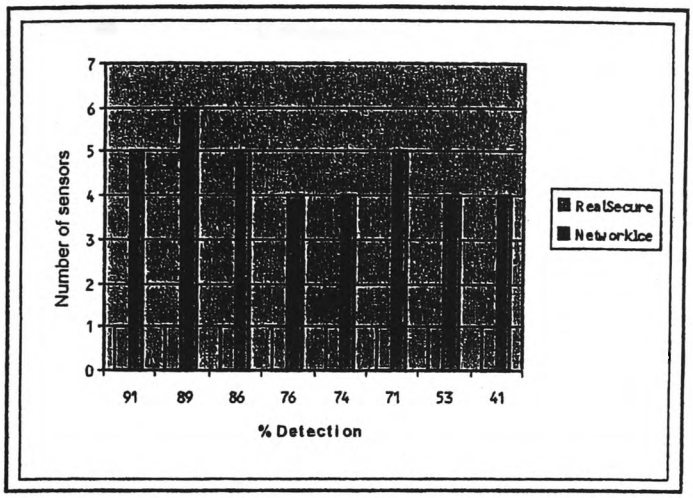


Fig. 8. % Detection by sensor type.

3. Even if the IDS knows what operating system every machine on the network runs, it still might not be able to tell just by looking at a packet whether a given machine will accept it. A machine that runs out of memory will discard incoming packets. The IDS has no easy way to determine whether this is the case on the end-system, and thus will assume that the end-system has accepted the packet. CPU exhaustion and network saturation at the end-system can cause the same problem.

Together, all these problems result in a situation where the IDS often simply can't determine the implications of a packet merely by examining it; it needs to know a great deal about the networking behavior of the end-systems that it's watching, as well as the traffic conditions of their network segments. Unfortunately, a network IDS doesn't have any simple way of informing itself about this; it obtains all its information from the packets it captures during attack detection.

4.6. Techno-economic analysis

The sole purpose of an intrusion detection system is to detect intrusions to the system it is protecting. But when choosing the right IDS product, IDS performance is not the only factor used in the selection process but some others such as scalability, availability and the total cost of the system relative to the price of the system the IDS is protecting, just to mention a few. That is why the overall evaluation of any IDS product is be based on a wide range of criteria. These criteria have been defined [19] in the common IDS architecture classification shown in Fig. 9. The architecture consists of both a quantitative and a qualitative component.

The following is a brief description of the classification.

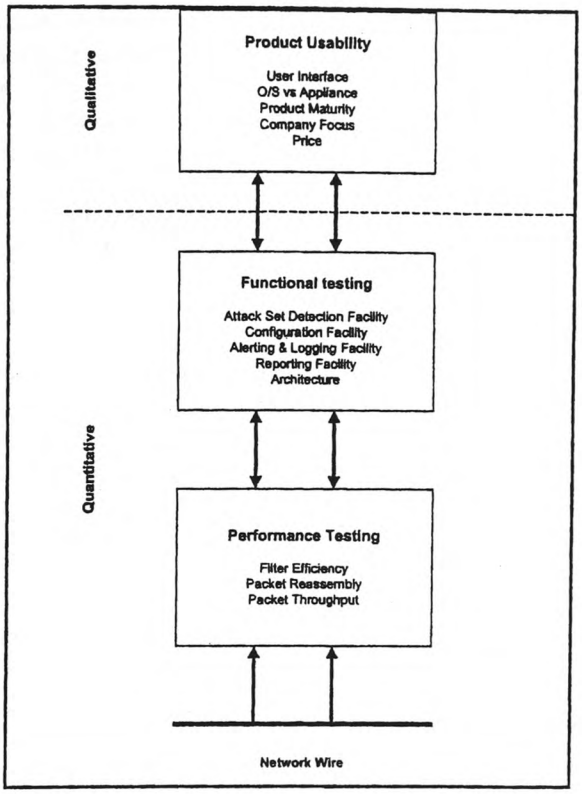


Fig. 9. Common IDS architecture [22].

4.6.1. Quantitative evaluation architecture

An IDS sensor's job is to watch the network and detect attacks a role that is performed by the packet-processing engine. To do this, the sensor looks at every packet on the network it is watching. The busier the network, the more packets there are to watch. If the sensor can't keep up, it will start to miss (or drop) packets. In the case that the attacks span multiple packets, the sensor holds the packets, assembles them and makes a determination on whether there is an attack. The extent and scope of accomplishing the above roles is the gauge of the effectiveness of the IDS and that is why the IDS performance is evaluated based on the ability of the processing engine effectively filter and reassemble packets to any given network throughput.

Equally important is the functionality of the IDS. In this case, the architecture is designed to define the operational setup that is used to assess the attack set detection, configuration alert triggering, logging and reporting facilities.

4.6.2. Qualitative evaluation architecture

The architecture defines the evaluation criteria of the IDS based on certain usability features such as ease of user interface (ease of use, ease of configuration, ease of filter customization); integration and interoperability with operating systems and existing network infrastructure; product maturity; company focus and price.

Although the focus of our work is primarily a subset of the Quantitative evaluation (performance testing), it is worth noting that there are other related issues that merit consideration. We have presented a comparison of the major features of both IDS products in Table 6.

From the perspective of cost benefit analysis, it is worthy to note that using our test results, 6 RealSecure sensors costing $53 970 will yield savings of almost $10 000 if deployment decision is based on the results on Fig. 6. It could also be said that the cost of the Gigabit sensor will fall with time thereby establishing an overall advantage over the use of multiple 100 Mbps sensors.

On the other major issues, both sensors in their numbers will trigger multiple alerts but the NetworkIce sensor will benefit from easier operational management with other collateral cost savings.

5. Implications for Gigabit Network Systems

The study presented in this paper provides a side-by-side comparison of two different techniques for intrusion detection. One being older (Megabit IDS) and the other representing evolutions from pure megabit IDS to gigabit IDS based on the extension of recurrent characteristics of ID system to new technologies.

The results are significant because the data on which the techniques are evaluated represent a significant corpus of empirically obtained data by which the probability of detection of a given intrusion detection technique can be simultaneously measured and evaluated against that of another technique in order to compare the correct detection rates that could aid the process of selecting the most feasible IDS product.

Given that Gigabit requirements will increasingly become mandatory especially for carrier networks with associated problems of information overload and data management, there is the need for commercial as well as corporate network infrastructures to tackle IDS issues that come along with this. For example, with the deployment of one Gbps IDS could come the issue excessive alerts triggers. This is even the case with the deployment of multiple NIDS devices, even at medium traffic rates that trigger so many alerts. Our tests were not intended to address all the issues associated with large-scale IDS deployment but the performance characteristics based on defined environmental settings, which we think could be a fairly good generalization. For instance, there are some common things on most networks – things like more TCP traffic than UDP traffic, packet size trends, a certain percentage of native fragmentation, etc.

Table 6
IDS features

Item	ISS NetworkIce	ISS RealSecure 5.5
Platform	Windows NT/2000	Solaris/Windows NT/2000
NIDS/HIDS agents	Y/n	Y/y
Integrated HIDS/NIDS management platform	N/a	Y
Integrates with file integrity checkers	N	Y
SNMP traps for integration into management platform	Y	Y
Back-end database API	Y	N
Management platform (console)	Web	Windows NT/2000
Remote sensor management	Windows NT/2000, Web	GUI
Stealth mode (unbound sniffing NIC)	Y	Y
Frag reassembly	Y	Y
TCP stream reassembly	Y	Y
Automatic signature update capabilities	N	Y
CVE cross-references	Y	N
Open signature rule sets	N	N
Customizable signatures	N	Y
Update frequency	As needed	Quarterly and mailing list alerts
Rule tuning (turn on/off specific signature)	Y	Y
Alerting mechanisms	E-mail, pager, SNMP, script	E-mail, OPSEC, TCP Kill, SNMP, blocking, log to database, alert to lucent firewall, paging, custom
Encrypted transmission upstream	Y	Y
Offending packet logging	Y	N
Standardized packet capturing	Y	N
Classification system	Info/serious/very serious/critical	Low/medium/high
24x7 support	Y	Y
Price	$63 000	Sensor: $8995

The point being made is that the study is more about trying to find a better technique to monitor networks against intrusion and not without context, arguing about the best NIDS which is about as useful as arguing about the best OS.

In this work, we believe that based on the facts that its important to note that the requirements of an enterprise network that is deploying a few devices locally to watch over a class-C is going to be different from that of a multi-national corporation that is deploying hundreds of devices.

Figures 5 and 6 show the performance of the equality matching the probability for detecting attacks at a particular threshold of network load. Generally speaking, if the threshold is set low, then the detection rate will be high. Similarly, a threshold set too high may end up not detecting most intrusions.

While the figures are useful for quickly determining how many attacks using a particular technique were detected, a more useful measure of the performance of the techniques can be obtained from analyzing the effects of the different parametric variables such as traffic type and size distribution and network conditions.

A measure of the overall effectiveness of a given intrusion detection system can be provided by the Probability of Detection (POD) curve. A POD curve is a parametric curve that is generated by varying the threshold of the intrusive measure (traffic type, particle size traffic distribution, etc.), which is a tunable parameter, and computing the probability of detection at each network utilization value. The curve is a plot of the likelihood that an intrusion is detected under defined network load conditions, against the likelihood that a non-intrusion is misclassified (i.e., a false positive) for a particular parameter, such as traffic type, particle size traffic distribution, tunable threshold, etc. The POD curve can be used to determine the performance of the ID system for different network load utilizations under any given configurable thresholds, or for comparing the performance of different intrusion detection techniques for given network utilization values.

5.1. Selective and adaptive deployment based on network capture/packet analysis

Packets per second and overall bandwidth are two of the common criteria for any networking equipment. These metrics are important for IDS for the same reasons. Packets per second allow quantification of the maximum amount of sustained load that can be handled with very little time to handle each packet. Maximum packet rate is achieved by lowering the packet size to the Ethernet minimum of 64 bytes. However, best practices entail pre-deployment analysis of the network traffic. This can be realized by testing the IDS over different packet sizes. Testing with only 64 byte packets, for sure, will make it difficult to evaluate the different effects variable packet sizes will have on the capture ability of the IDS sensor. As a result, this will not be effective.

From the performance standpoint, NIDS observes packets on the wire. If packets are sent faster than the NIDS can process them, there is no degradation in the network performance because the NIDS does not sit directly in the flows of data. However, the NIDS will lose effectiveness and packets could be missed causing both false-negatives and false-positives. It is therefore better to avoid exceeding the capabilities of IDS so as to maximize benefits. From a routing standpoint, IDS, like many state-aware engines, does not operate properly in an asymmetrically routed environment. Packets sent out from one set of routers and switches and returning through another will cause the IDS systems to see only half of the traffic, causing false-positives and negatives.

An IDS is most useful when there is an unusual traffic stream passing through the network. If the IDS could not handle the full bandwidth available with an upstream provider, at 64 byte packets, and a denial of service (DDoS) attack comprised of only 64 byte packet is initiated, the IDS will fail. There is therefore a need to have an IDS that can handle 64 byte packets up to and including the full load that could possibly be sent it's way. At the same time, the IDS needs to still watch and alarm at activity that might be going on in the 'noise' of the DDoS.

6. Conclusion

In this paper we have shown how a probabilistic method can be used to determine the performance of the IDS in a Gigabit traffic stream for multiple IDS 100 Mbps sensors. Two misuse detection techniques for evaluating the performance of the IDS against tunable parametric specifications were presented. The techniques use a uniform methodology to measure the ability of the IDS to detect attacks under varying test parameters. The ability of intrusion detection systems to correctly identify the attacks was measured under different network configurations. The performance of the different intrusion detection systems was compared by testing them with known common vulnerability exploits (CVE) provided by the SAN Intrusion Detection Evaluation program. The results of this analysis provide us with a probabilistic framework for assessment of deployment of IDS's within a Gigabit traffic stream.

In this regard, the test results show that in general, IDS performance is greatly influenced by bandwidth utilization. The following are conclusions drawn from the study:

1. For a Gigabit traffic throughput ranging from 100 to 600 Mbps, more than 75% attack detection rate is realizable with a maximum of 6 RealSecure sensors or 1 NetworkIce sensor. Beyond this point, detection rate drops to undesirable levels although only about 4 RealSecure sensors will be needed to match the detection rate of 1 NetworkIce Gigabit sensor.
2. NetworkIce sensor is more effective in detecting attacks in the low Gigabit level range (100 to 300 Mbps), while multiple Realsecure sensors are more effective at Gigabit traffic throughput ranging above 300 Mbps.
3. Generally, the detection rate for both types of IDS decreases with increase in network utilization. The decrease is more pronounced above 600 Mbps of network utilization.
4. Additionally, beyond 800 Mbps the detection rate falls to undesirable limits for both IDS sensor types.

Finally, it is concluded that currently available IDS products if selectively utilized based on effective deployment techniques are realistic technologies that could provide a reasonable measure of security monitoring in Gigabit networks with large

traffic volumes. In comparative terms, when the cost and other techno-economic factors are taken into account, the use of a single Gigabit IDS sensor instead of multiple 100Mbps IDS sensors will be advantageous.

Acknowledgement

The authors are grateful to anonymous reviewers for providing constructive comments on a previous version of the paper. The support provided by the following is gratefully acknowledged: Ramiz Mansour, Internet Security Systems, Inc. and Clark McDaniel, Network Associates, Inc.

Appendix

Specifications of the SmartBits ML-7710

The 10/100 Ethernet SmartBit cards (ML-7710) are interfaces that transmit traffic to the various interfaces of devices under tests. It is used for network performance analysis for 10/100/Gigabit Ethernet, ATM, Packet over SONET, Frame Relay, xDSL, Cable Modem, IP QoS, VoIP, Routing, MulticastIP, and TCP/IP.

Specifications of the AppSwitch AS3502

The AppSwitch is a switch that was designed to include IDS redundancy, flood protection and 100% inspection of attacks. Data can be gigabit broken into smaller 100 mbps chunks or many 100 mbps segments combined into a gigabit IDS (and combinations between). The following are the specifications.
One 1000Base-SX input port
12 10/100Base-TX full duplex output ports
Processor: 125 Mhz RISC
RAM: 128 Mb DRAM/4 Mb SRAM
Firmware/OS version 3.11

Attacker configuration

The attacker is an Intel-based system running Windows NT 4.0 server (with Service Pack 6) loaded with Network Associates CyberCop scanner which has a modular IDS test suite and interactive CASL script generator.

Target system configuration

The targets are two Windows NT 4.0 servers with Microsoft IIS Web server, ftp server, and Hermes mail server software installed and active.

Background traffic generator

The load generator was a SmartBits tester configured to generate 3 UDP sessions from each of the 12 transmitting ML7710 modules for a total of 36 Sessions or streams.

RealSecure Sensor

A standard PC with the following hard and software configuration: Pentium III 600 MHz processor, 256 Mb RAM, 100 Mb disk space plus 100 Mb per managed sensor on the console. NT 4.0 Workstation with SP6, Internet explorer 5.5 and two 100 Mbps Ethernet Intel Pro/100+ PCI adapters (one configured for stealth monitoring and the other in promiscuous mode). The sensor analyzes the packets on the wire and alerts if it senses an attack.

NetworkICE Gigabit sensor

NetworkICE Gigabit sensor was configured per manufacturer's specification.

References

[1] D. Anderson, T. Frivold and A. Valdes, Next-generation intrusion–detection expert system (NIDES): Final technical report, Technical report, Computer Science Laboratory, SRI International, Menlo Park, CA, 16 November 1994.

[2] K. Claffy and S. McCreary, Internet measurement and data analysis: Passive and active measurement, in: *Proceedings of American Statistical Association*, New Jersey, 1999.

[3] D.E. Denning, An intrusion-detection model, *IEEE Transactions on Software Engineering* 13(2), (1987).

[4] B. Guha and B. Mukherjee, Network security via reverse engineering of TCP code, in: *Proceedings of the IEEE Infocom-96 Conference*, 1996.

[5] L.T. Heberlein, G. Dias, K.N. Levitt, B. Mukherjee, J. Wood and D. Wolber, A network security monitor, in: *Proceedings of the 1990 Symposium on Rematch in Security and Privacy*, Oakland, CA, IEEE Computer Society, 1990, pp. 296–303.

[6] J. Kadambi et al., *Gigabit Ethernet – Migrating to High-Bandwidth LANS*, Prentice Hall, New Jersey, 1998, pp. 167–234.

[7] C. Iheagwara and A. Blyth, Evaluation of the performance of IDS systems in a switched and distributed environment, *The International Journal of Computers and Telecommunications Networking*, Elsevier, London (in press).

[8] C. Iheagwara and H. Harding, Experiences in the operation of the RealSecure network intrusion-detection system (NIDS), Technical report, Una Telecom, Inc. Silver Spring, MD, September 1999.

[9] G. Jakobson and M.D. Weissman, Alarm correlation, *IEEE Network* (November) (1993), 52–59.

[10] K. Jackson, D. DuBois and C. Stallings, An expert system application for network intrusion detection, in: *Proceedings of the Fourteenth Computer Security Group Conference*, Department of Energy, 1991.

[11] S. Kliger, S. Yemini, Y. Yemini, D. Ohsie and S. Stolfo, A coding approach to event correlation, in: *Proceedings of the Fourth International Symposium on Integrated Network Management (IFIP/IEEE)*, Santa Barbara, CA, Chapman and Hall, London, England, 1995, pp. 266–277.

[12] T.F. Lunt, R. Jagannathan, R. Lee, A. Whitehurst and S. Listgarten, Knowledge-based intrusion detection, in: *Proceedings of the 1989 AI Systems in Government Conference*, 1989.

[13] T.F. Lunt, A. Tamaru, F. Gilham, R. Jagannathan, C. Jalali, P.G. Neumann, H.S. Javitz and A. Valdes, A real-time intrusion-detection expert system (IDES), Technical report, Computer Science Laboratory, SRI International, Menlo Park, CA, 28 February 1992.

[14] U. Maimon, Port scanning without the SYN flag, *Phrack Magazine* 7(49) (1998).

[15] M. Mansouri-Samani and M. Sloman, Monitoring distributed systems, *IEEE Network* (November) (1993), pp. 20–30.

[16] K. Meyer, M. Erlinger, J. Betser, C. Sunshine, G. Goldszmidt and Y. Yemini, Decentralizing control and intelligence in network management, in: *Proceedings of the Fourth International Symposium on Integrated Network Management (IFIP/IEEE)*, Santa Barbara, CA, Chapman and Hall, London, England, 1995, pp. 4–16.

[17] R. Morris, A weakness in the 4.2BSD UNIX TCP/IP software, in: *Computing Science Technical Report 117*, AT&T Bell Laboratories, Murray Hills, NJ, 25 February 1985.

[18] A. Mounji, B. Le Charlier and D. Zampunieris, Distributed audit trail analysis, in: *Proceedings of the ISOC 1995 Symposium on Network and Distributed System Security*, 1995, pp. 102–112.

[19] P.A. Porras, STAT: A State Transition Analysis Tool for intrusion detection, Master's thesis, Computer Science Department, University of California, Santa Barbara, July 1992.

[20] P.A. Porras and A. Valdes, Live traffic analysis of TCP/IP gateways, in: *Internet Society's Networks and Distributed Systems Security Symposium*, March 1998.

[21] J. Postel, Internet protocol, request for comment, RFC 791, Technical report, Information Sciences Institute, September 1981.

[22] K. Richards, Network Based Intrusion Detection: a review of technologies, *Computers & Security* 18 (1999), 671–682.

[23] S. Staniford-Chen and L.T. Heberlein, Holding intruders accountable on the Internet, in: *Proceedings of the IEEE Symposium on Security and Privacy*, 1995.

[24] S.R. Snapp, J. Brentano, G.V. Dias, T.L. Goan, L.T. Heberlein, C.L. Ho, K.N. Levitt, B. Mukherjee, S. Smaha, T. Grance, D.M. Teal and D. Mansur, DIDS (Distributed Intrusion Detection System) – motivation, architecture, and an early prototype, in: *Proceedings of the Fourteenth National Computer Security Conference*, Washington, DC, NIST/NCSC, 1991, pp. 167–176.

[25] C. Spurgeon, Ethernet: The definitive Guide, February 2000, p. 340.

[26] W. Venema, Project SATAN: UNIX/internet security, in: *Proceedings of the COMPSEC-95 Conference*, Elsevier, London, 1995.

[27] http://www.silicondefense.com/software/acbm/speed_of_snort_03_16_2001.pdf

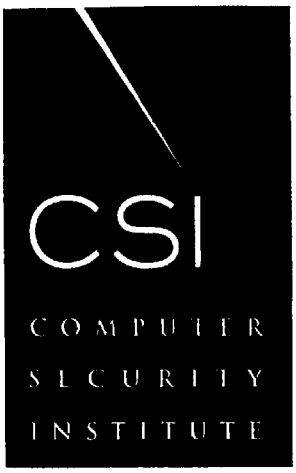
[28] http://public.lanl.gov/mfisk/papers/ucsd-tr-cs2001-0670.pdf

[29] http://www.caida.org/outreach/papers/int98

Printed in The Netherlands

Appendix 6

"Towards an Effective Risk Assessment Methodology: Factoring in Novel Concepts For Assessing Intrusion Detection Systems in Complex Infrastructures" Computer Security Journal, Volume XIV, Number 2, 2003



COMPUTER SECURITY JOURNAL

VOLUME XIX, NUMBER 2, SPRING 2003

Economics:

Information Security Expenditures and Real Options

A wait-and-see approach—page 1

Risk Assessment:

More Effective Risk Assessment

Charles Iheagwara uses cascading threat multipliers to assess intrusion detection systems—page 8

Issues & Trends:

CSI/FBI 2003 Computer Crime and Security Survey

The eighth annual survey shows costs are down—page 21

Counterpoint:

What Does the CSI/FBI Survey Really Tell Us?

What else should the survey ask?—page 41

More Effective Risk Assessment

Using Cascading Threat Multipliers for Assessing Intrusion Detection Systems in Complex Infrastructures

by Charles Iheagwara, P.E., CCNP, GAIC, MCSE

1.0 Abstract

This paper discusses emerging approaches to risk assessments, as well as the issues and challenges that are presented when applying certain risk assessment methods to increasingly complex and interdependent infrastructures, such as government agencies with dispersed locations and distributed architectures. Further, the paper provides an approach to support decision-making about appropriate techniques for the assessment of the IDS or other security products in a given networked system. On a broader level, it is hoped that the novel concept presented here will serve as a foundation for further developing and formalizing the description, asset valuation, analysis and selection of tools to support risk management of security products in complex environments.

2.0 Introduction

The constantly evolving and fast-changing corporate security policies and government regulations aimed at addressing the increasingly networked environment and complex of interdependent infrastructures, risk management is becoming an increasingly important tool in corporate management strategies.

"Financial losses to business and government due to Internet vulnerabilities could exceed $100 billion per year by 2004" [DOS01]. It seems to be a realistic view, since *Computer Economics*, a California-based Internet research organization, estimates the economic impact of only the last four major malicious code incidents (Love Bug, SirCam, Code Red, Nimda) to be over $13 billion [COM02]. It is obvious to say that "as the number of companies conducting business on the Internet rises, so too does the sophistication and number of cyber attacks."

Based on their own projections, *Computer Economics* notes that the probability of targeting and hitting each organization is growing: "Computer crime will grow by an estimated 230 percent during 2002. Similar trends are expected with Internet fraud, which will be up over 100 percent, and viruses, which will increase by 22 percent during the same period." Even more disturbing is underreporting: "According to government and industry sources, only about 20 percent of computer security violations are actually reported" [COM02].

Often risk assessments are conducted as a result of mandatory government audit requirement. For example, the U.S. GAO, as a result of an annual audit of a selected government agency, reports that the information system controls is a material weakness in the agency's Highway Trust Fund's financial audit report for the fiscal year under review. This could then prompt an audit of the information system general controls.

The GAO evaluation is usually based on the Federal Information System Controls Audit Manual (FISCAM), which contains guidance for reviewing information system controls that affect the integrity, confidentiality and availability of computerized data, along with the May 1998 GAO study of security

management best practices at leading organizations.

As a result of the GAO audit, the agency's management could conduct a risk assessment on its IT infrastructure, including critical information system control mechanisms, local and wide area networks. The assessment will serve as the agency's critical business driver for IT security practice and any subsequent mitigation and compliance program.

In the past decade, several risk management approaches have been introduced [RISKIT] and while some organizations, especially in the U.S. defense sector [RISKIT], have defined their own risk management approaches, most organizations do not manage their risks explicitly and systematically. Risk management based on intuition and individual initiative alone is seldom effective and rarely consistent either across time or across industry norms.

The use of risk management in corporate entities has not been without problems. There are various reasons for this, including known issues with user acceptance of the results. When risk management methods are used, they are often simplistic and users have little confidence in the results of their risk analysis results. The following factors contribute to the low use of risk management methods in practice:

- Risk is an abstract and fuzzy concept and users lack the necessary tools to define risk more accurately for deeper analysis.
- Many current risk management methods are based on quantification of risks for analysis and users are rarely able to provide sufficiently accurate estimates for probability and loss for the analysis results to be reliable. On the other hand, table-based approaches are often biased and too coarse for risk prioritization.
- Risks have different implications to different stakeholders. Few existing methods provide support for dealing with these different stakeholders and their expectations.
- Each risk may affect a project in more than one way. Most existing risk management approaches focus on cost, schedule or quality risks, yet their combinations or even other characteristics (such as future maintenance effort or company reputation) may be important factors that influence the real decision-making process.
- Many current risk management methods are perceived as too complex or costly to use. A risk management method should be easy to use and require a limited amount of time to produce results; otherwise it will not be used.

Given the increasing interest in risk management, there is the need for the present risk management methods to gain wider acceptance. This can be realized by evolving effective strategies to address the aforementioned issues. Furthermore, risk management methods should also provide comprehensive support for risk management in projects, practical guidelines for application, reasons for communications between participants and credibility.

3.0 The Basic Principles of Effective Risk Assessment

An effective risk assessment method should be able to address the issues (factors that contribute to the low usage of risk management methods in practice) listed. The specific characteristics can be described by the following principles [RISKIT].

1. The risk assessment method should result in explicit definition of objectives, constraints and other drivers that influence the project. Risk is a relative concept; its definition depends on expectations that are associated with a situation. In order to analyze risks, it is necessary to formalize the expectations. When expectations are recognized and defined, we refer to them as goals. While some goals cannot be stated precisely, at least they should be identified and documented as well as the available information allows. The method contains an explicit step and supporting templates to assist in the goal definition.

2. The risk assessment method should provide precise and unambiguous definitions for risks. The common definition of risks, either by dictionaries or everyday usage, associates several different meanings to risk. It can refer to a possibility of loss, the actual loss that would result if the risk occurs, a factor or element that is associated with a threat, or a person that contributes to the possibility of loss [RISKIT]. The dictionary definitions for risk are so broad that it is fair to define risk as anything that is related to the possibility of loss. Clearly, there is some value in having such a broad and encompassing concept to facilitate initial discussion about risk. However, we believe that this wide range of meanings associated to the word "risk" can also prevent adequate precision in more detailed

analysis or risks unless this ambiguity is explicitly addressed and removed.

3. The risk assessment method should be aimed at modeling and documenting risks qualitatively. The method provides conceptual and graphical tools to model different aspects of risks qualitatively, instead of requiring quantitative estimation of risk probability and impact to take place early in the project. Given the difficulty of these estimations and the often-ambiguous interpretations of risks, the margin of error in risk quantification is high. By emphasizing the qualitative understanding of risks, there is a better basis for understanding and communicating risk.

4. The risk assessment method should be able to use both ratio and ordinal scale risk ranking information to prioritize risks reliably. The method should reduce the estimation problem. Instead of forcing the quantification of risks using ratio scale metrics (often an unrealistic goal), the method only attempts to accomplish the necessary quantification of risks. For risk management purposes, it may be enough to identify the biggest risks and propose action to control them, while the exact values of probability and loss may not be needed. The selection of the type of metrics to be used in risk analysis should be based on the objectives of the analysis and the availability of data about risks.

5. The risk assessment method should use the concept of utility loss to rank the loss associated with risk. Many current risk management approaches are based on ranking of risks—based on the loss they cause to some specific attributes of the project, such as cost, time delay or quality metrics. Often a single metric is used. This can be detrimental for two reasons. First, the use of a single metric, or a small number of metrics, can create strong bias away from secondary, yet influential goals that should be considered. Second, research in economics and management science has strongly indicated that decisions are made based on the changes in the expected utility (or utility loss) of alternatives. As the utility functions of stakeholders are likely to be non-linear, use of direct loss metrics can lead to wrong estimates and rankings of the risks. Therefore, the risk assessment method should use the concept of utility loss to compare and rank losses of risks.

6. Different stakeholder perspectives should be explicitly modeled in the risk assessment method. All projects have more than one stakeholder that is interested in its results. They may have different priorities and levels of expectations. Risk management should be based on the recognition of these stakeholder expectations and priorities. Traditionally, direct project metric based approaches cannot easily support the comparison of different stakeholder views and few risk management approaches attempt to address the issue. The risk assessment method should support stakeholder views by documenting their expectations explicitly and evaluating the utility loss for each separately.

7. The risk assessment method should have an operational definition and training support. The risk assessment method should have an operational definition so that it can be applied easily and consistently. There should be a tutorial available and an application guideline.

4.0 Definitions and Frameworks in Risk Management

Risk, the possibility of damage or loss, is described mostly in dependencies of threat and vulnerability, or impact and probability. The general framework developed by NIST workshops in 1992 [CRA98] formalized six concepts in risk analysis: *"assets, vulnerabilities, threats, impacts, likelihoods and safeguards."* In another framework Ozier lists 12 elements of risk by indicating quantifications and dependencies (e.g. motivation, capability and resource availability for threat agents) for some of them [OZI99].

There is a wide consensus among information security professionals that a 100 percent infallible security solution is not realistic or affordable. Failsafe security plans are often not practical since the measures would cost more than the asset value to be protected. Thus the emphasis of dealing with risks in this context moves from risk avoidance to risk management. Basically risk analysis and risk management are defined as follows [CHI97]:

- Risk analysis involves the identification and assessment of the levels of risks calculated from the known values of assets and the levels of

threats to, and vulnerabilities of, those assets.

- Risk management involves the identification, selection and adoption of countermeasures justified by the identified risks to assets and the reduction of those risks to acceptable levels.

Thus, the measure of risk can be determined as a product of threat, vulnerability and asset values.

The risk elements and their corresponding countermeasures can best be visualized with a cuboid (Figure 1). The system has an initial level of risk before any countermeasures are applied. Countermeasures, assuming that their values are assigned by the same parameters that are used for threat, vulnerability and asset valuation, can reduce risk, i.e., by reducing threat (locked doors, firewalls), reducing vulnerability (awareness, patches, hot fixes) or reducing asset value (encryption). After calculating the results from each combination of threat, vulnerability, asset and countermeasure the residual risk is determined [BRE00]. Here the impact element is covered in asset value, the likelihood in threat and vulnerability values.

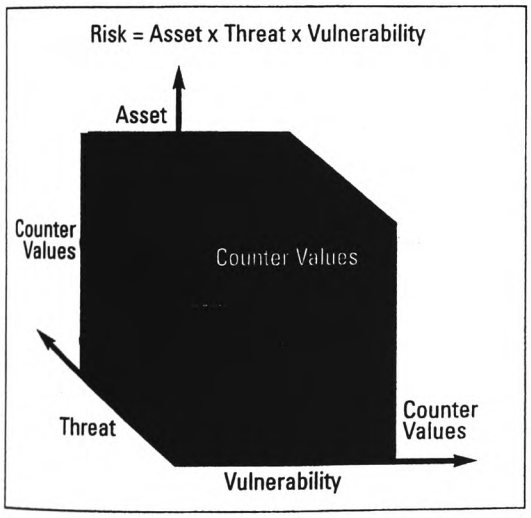


Figure 1 **Risk as a function of asset value, threat**

Considering the business environment and resources available, the decision-makers in an organization may then implement one or more of the following risk management strategies:

- Risk mitigation (reducing the risks with applying selected countermeasures)
- Risk acceptance (accepting the residual risk or even the initial level if the countermeasures are more costly than the asset values)
- Risk transfer (transferring the risk to another organization, e.g., by insurance or outsourcing)

The option of eliminating assets may also be mentioned here in case of very high risk, unavailable or unaffordable countermeasures as well as impossible risk transfer.

5.0 Risk Assessment Methodologies

5.1 Development phases

The present theory of risk assessment is relatively robust because of significant developments flowing from first- to fourth-generation tools. While the concept of risk has not changed very much, the approach and the technology have. At the present time we can categorize risk assessment methodology (technology) as belonging to four generations. The first consists of paper-based methods, e.g., CRAMM, "Memo 10." These methods deal with risk in simple terms (high, medium and low), they are ignorant of specific software vulnerabilities, deal in generic network topologies and use look-up tables in order to calculate the risk. Second-generation tools are merely a software version of their first-generation counterparts. Third-generation tools (such as "Expert") make use of vulnerability-safeguard libraries that are regularly updated, which allow for network scanning and the use of more sophisticated algorithms. Fourth-generation tools (e.g., RiskWatch) have undergone considerable refinements that make it possible for the effectiveness of safeguards to be determined and the risk of different networks to be compared.

Today, risk analysis methodologies include identification and valuation of assets, followed by identifying threats likely to occur to them with related vulnerabilities. Finally, risk is determined for combinations of identified assets, threats and vulnerabilities to propose appropriate countermeasures. During this process two different measurement schemes can be applied to risk elements; quantitative or qualitative. The quantitative approach articulates risk in numerical terms, i.e., expected monetary loss and probability (e.g., Annual Loss Expectancy, ALE). The qualitative approach has no numeric

value and is usually opinion based. Results are summarized in words like "low," "medium" and "high."

There are several methodologies promulgated via U.S. government FIPS 65 guideline (withdrawn in 1995) for performing risk analysis in large data processing centers [GIL89]. Recent approaches attempt to adapt technological advances like the Internet by prototyping real-time risk analysis [VEN99] and emerging applications like e-commerce by using case-based reasoning [CHA99] or consider a framework for the "whole system" during the risk management life cycle [CRA98]. In the latter work and especially in [LAB99] three generations of risk analysis and management methodologies are identified and their shortcomings discussed, in which the first generation corresponds to the mainframe era, the second to networks and distributed computing, and the third to open environments and the Internet.

If a security level is high and a threat frequency is low, it is hard to assign corresponding numbers to these ratings and come up with a useful outcome.

5.2 Approaches/Techniques

Quantitative and qualitative risk analyses offer two perspectives to the assessment of risk. The argument for justifying quantitative risk assessment is that cost-effective safeguards cannot be evaluated against losses unless the risks are quantified. Qualitative methodologies emphasize descriptions rather than calculations. Quantitative risk assessments make use of a mathematical calculation produced from the probability of an event occurring and the likely loss should it occur. This is called the Annual Loss Expectancy (ALE). Probability can rarely be precise and can, in some cases, promote complacency. In addition, controls and countermeasures often tackle a number of potential events and the events themselves are frequently interrelated. In qualitative risk assessment, probability data is not required and only estimated potential loss is used. The quantitative approach attempts to assign real numbers to the costs of countermeasures and the amount of damage that can take place [Harris]. The quantitative approach also provides concrete probability percentages when determining the likelihood of threats and risks. Each element within the analysis (asset value, threat frequency, severity of vulnerability items) is quantified and entered into equations to determine total and residual risks. Purely quantitative risk analysis is not possible because the method is attempting to quantify qualitative items. If a security level is high and a threat frequency is low, it is hard to assign corresponding numbers to these ratings and come up with a useful outcome.

The classic quantitative algorithm, as presented in FIPSPUB-65, which laid the foundation for infor-

Table 1 **Example of a Qualitative Analysis (Harris)**

	Severity of Threat	Probability of Threat Taking Place	Potential Loss to the Company	Effectiveness of Firewall	Effectiveness of Intrusion Detection System	Effectiveness of Honeypot
IT Manager	4	2	4	4	3	2
Database Administrator	4	4	4	3	4	1
Application Programmer	2	3	3	4	2	1
System Operator	3	4	3	4	2	1
Operational Manager	5	4	4	4	4	2
Results	3.6	3.4	3.6	3.8	3	1.4

mation security risk assessment, is presented below:

(Asset Value x Exposure Factor = Single Loss Expectancy) x Annualized Rate of Occurrence = Annualized Loss Expectancy

For example, let's look at the risk of fire. Assume the Asset Value is $1 million, the exposure factor is 50 percent, and the Annualized Rate of Occurrence is 1/10 (once in 10 years). Plugging these values into the algorithm yields the following:

($1M x 50% = $500K) x 1/10 = $50K

Using conventional cost/benefit assessment, the $50K ALE represents the cost/benefit break-even point for risk mitigation measures. In other words, the organization could justify spending up to $50K per year to prevent the occurrence or reduce the impact of a fire.

The qualitative approach does not assign monetary values to components or losses. Instead, qualitative methods walk through different scenarios of risk possibilities and rank the seriousness of the threats and the sensitivity of the assets. Qualitative analysis techniques include judgment, intuition and experience. Examples of qualitative techniques are Delphi, brainstorming, storyboarding, focus groups, surveys, questionnaires, checklists, one-on-one meetings and interviews.

Because of the nature of risk assessment, there are inherent advantages and disadvantages in both methodologies.

5.3 Quantitative vs. Qualitative

In the following brief analysis, the features of specific risk assessment tools will not be discussed. Rather, the pros and cons associated in general with qualitative and quantitative methodologies will be addressed.

Quantitative and qualitative approaches have their own pros and cons and each applies more appropriately to certain situations. The organization, risk analysis team, and the tools they decide to use will determine which approach is best based on the culture and local environment. There are several advantages and disadvantages for each assessment methodology.

Qualitative—advantages

- ❒ Calculations, if any, are simple and readily understood and executed.
- ❒ It is usually not necessary to determine the monetary value of information (its availability, confidentiality and integrity).
- ❒ It is not necessary to determine quantitative threat frequency and impact data.
- ❒ It is not necessary to estimate the cost of recommended risk mitigation measures and calculate cost/benefit.
- ❒ A general indication of significant areas of risk that should be addressed is provided.

Qualitative—disadvantages

- ❒ The risk assessment and results are essentially subjective in both process and metrics. The use of independently objective metrics is eschewed.
- ❒ No effort is made to develop an objective monetary basis for the value of targeted information assets. Hence, the perception of value may not realistically reflect actual value at risk.
- ❒ No basis is provided for cost/benefit analysis of risk mitigation measures, only subjective indication of a problem.
- ❒ It is not possible to track risk management performance objectively when all measures are subjective.

Quantitative—advantages

- ❒ The assessment and results are based substantially on independently objective processes and metrics. Thus meaningful statistical analysis is supported.
- ❒ The value of information (availability, confidentiality and integrity), as expressed in monetary terms with supporting rationale, is better understood. Thus, the basis for expected loss is better understood.
- ❒ A credible basis for cost/benefit assessment of risk mitigation measures is provided. Thus, information security budget decision-making is supported.
- ❒ Risk management performance can be tracked and evaluated.
- ❒ Risk assessment results are derived and expressed in management's language, monetary value, percentages and probability annualized. Thus risk is better understood.

Quantitative—disadvantages

- ❒ Calculations are complex. If they are not understood or effectively explained, management may mistrust the results of "black box" calculations.
- ❒ It is not practical to attempt to execute a quantitative risk assessment without using a recognized automated tool and associated knowledge bases.

A manual effort—even with the support of a spreadsheet and generic statistical software—can easily take 10 to 20 times the work effort required with the support of a good automated risk assessment tool.

- ❒ A substantial amount of information about the target information and its IT environment must be gathered.
- ❒ As of this writing, there is not yet a standard, independently developed, and maintained threat population and threat frequency knowledge base. Thus users must rely on the credibility of the vendors who develop and support extant automated tools or do threat research on their own.

What is known with certainty is that all the methods involve some level of subjectivity and the quantitative approach expresses risks in monetary values. Quantitative assessments are ineffective in organizations with poor IT accounting management practices. This is because of the high dependence of the automated tool on data input accuracy. When and where that is the case, the qualitative methodology should be the appropriate approach.

Since the qualitative metrics are all subjective in nature, the first two metrics, "Low, Medium and High, or Ordinal Ranking," can characterize virtually every risk element. "Vital, Critical and Important," however, are descriptive only of an asset's value to an organization.

The qualitative approach makes no effort to scale risk or to value information assets. Rather, the approach seeks to identify in-place safeguards, compare those with what industry peers are doing to secure their information, and then enhance security wherever it falls short of industry-peer security. A further word of caution is appropriate here. The approach is founded on an interpretation of "due care" that could sometimes be at odds with the well-established legal definition of "due care."

6.0 Risk Assessment of IDS/ Security Products in Complex Environments

6.1 Complex environments

Today's technology base is becoming increasingly large and complex. Networks are growing, and applications are being migrated from centralized systems to client-server environments. In addition, organizations are connecting their networks to those of other organizations and to the Internet at a rapid rate. All of this added complexity presents a challenge to administrators who are responsible for managing these systems. The growth in the number of networked systems has increased their complexity and has raised the threshold of expertise required of these administrators.

Complexity has likewise increased due to the desire to integrate the operational data of an enterprise and to provide centralized (thus controlled) access to that data. The technology of computer networks, on the other hand, promotes a mode of work that goes against all centralized efforts.

The enterprise network is a system that interconnects a multitude of computers and devices for communications and information/resource sharing. The design of an enterprise network is often an assembly of very dissimilar components. To keep the various interconnected parts of the system interoperable, appropriate data transport and exchange technologies, rules and procedures are implemented.

The complexity of systems consisting of independent, interacting components lies in the extremely large number of possible factors in which actions of the individual components can interleave.

The network design often incorporates an outside network, an intermediary (DMZ) network and an internal network. Among the mission-critical servers/devices in the network system are VPN, Web, FTP, DNS/Mail servers, routers, and PIX firewalls with redundant failover mechanisms and local directors for load balancing. A typical topology of the network is shown in Figure 2 (following page).

Thus, the complexities range from different systems applications to varied complex topological designs within a cross-functional business environment. Because of these complexities, it is often difficult to conduct accurate asset valuation of the individual components within the system.

6.2 Difficulties of asset valuation of networked devices in complex environments

Before appropriate security tools can be identified, valued and assessed, an organization must conduct a comprehensive examination of its networked assets. The organization must know and have documented all of its current and anticipated information assets,

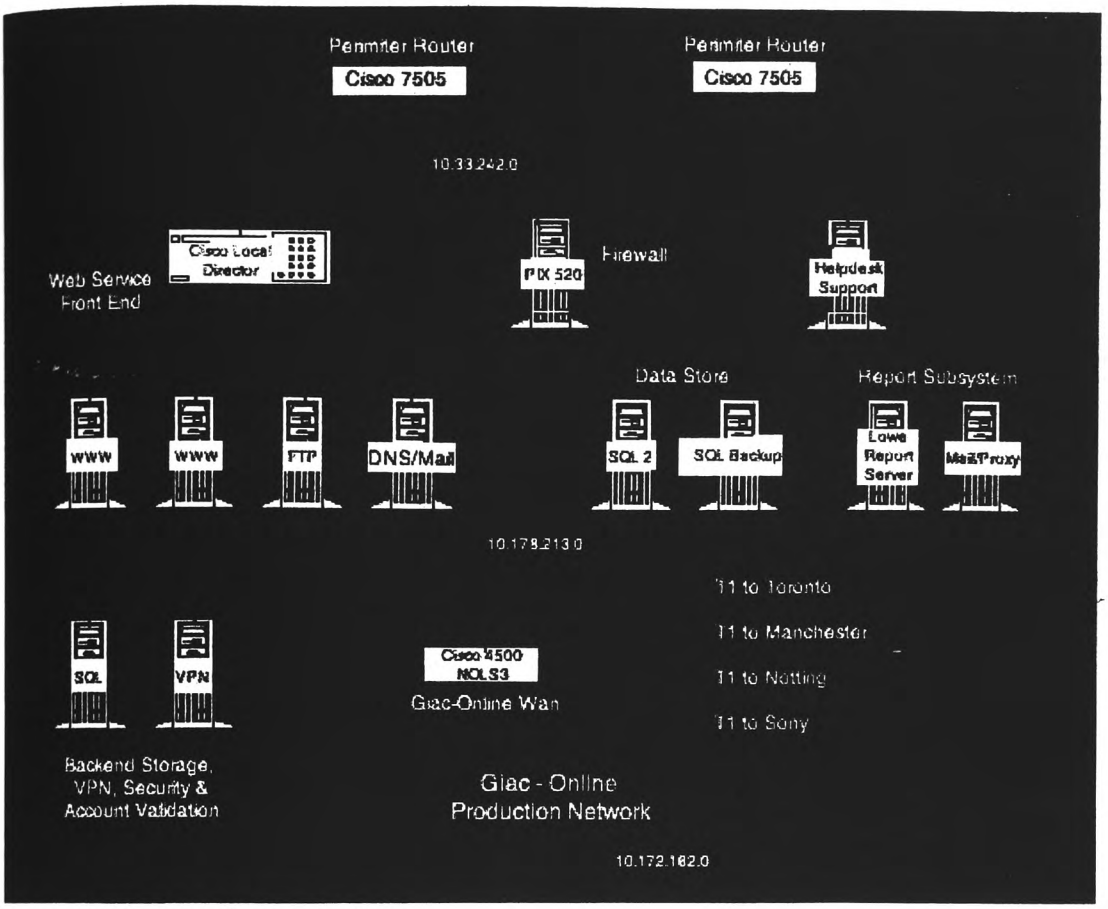


Figure 2 **Typical enterprise network**

along with the infrastructure in which these assets are stored and communicated. Policies must be clearly defined regarding expectations and responsibilities for physical, personnel and networked systems security, as well as the relationships between them. The organization must define expectations about how information is communicated between internal and external entities. As the needs and activities of an organization change with time, it is imperative that knowledge about the organization's information assets, infrastructure, personnel and policies be kept up-to-date and consistent with one another. The organization must also maintain current knowledge about the kinds of security problems to which their information assets, infrastructure and personnel may be susceptible. Once these preparations have been completed, and procedures put in place to maintain the currency and accuracy of policies and knowledge of the organization, then specific security objectives may be defined for each information asset and service.

The practical reality is that the tasks involved in risk assessments can be overwhelming. This is because the process tends to establish metrics as yardsticks to proffer remedies while at the same time trying to maintain, to a considerable degree, a high level of accuracy.

In complex environments, conducting risk assessments becomes even more complex as a result of the difficulties created by the interplay of people, technology and operations. Risk determination even with the simplest asset could be complex.

Historically, the use of automated tools has alleviated these problems in blunt fashions, but they can be used successfully in environments where there are good accounting practices and where the outcome of automated report satisfies the requirements of the information system management. However, it does seem that automated tools generally cannot meet all the requirements of complex, multi-dimensional environments with non-traditional or governmental accounting practices that require extensive modification.

The determination of the asset value when there is interdependence in networked environments could be extremely difficult. This is because of the asset value when taken in up- and downstream dimensions. Thus, an asset such as IDS can be measured in so many dimensions and in tangible and intangible measures.

During risk assessments, certain challenges could be encountered, especially in complex infrastructures where the valuation of mission-critical security equipment like IDS could be difficult. Other difficulties could manifest in obtaining the necessary data input needed for the quantitative calculations when the quantitative technique is used. The associated difficulties include:

- ❒ Inability to obtain input data when the agency's accounting practices do not readily provide the type of asset value data required by automated quantitative assessment tools.
- ❒ Difficulties in obtaining meaningful financial information. Subsequently, inadequate financial data invalidates single and annualized loss expectancy (SLE and ALE) calculations, and improperly skews recommendations for safeguards with the greatest return on investment (ROI).
- ❒ Lack of clear definitions for the agency's organizational structures, which contribute to the challenge of not being able to place dollar values on assets. This affects asset values for personnel and supporting infrastructure.
- ❒ Inability to obtain the agency's share of responsibilities with other intra-agency entities—which makes it very difficult to assign partial responsibility.

Because of the importance of the valuation process, I propose to introduce a new approach for risk assessment asset valuation in complex infrastructures using the IDS as an illustration.

6.3 Novel concepts for effective IDS risk assessment

The main concepts of risk management and related equations are given in Table 2 (opposite page). In the world of new technologies, the interplay of technological processes, policies and risk management introduces complexities into risk management and requires the development and introduction of new approaches and concepts like that of the cascading threat multiplier (CTM) [KEV02] to accurately conduct valuation studies or calculate ROI for any acquired or developed technology.

CTM factors in the importance of other critical assets tied (networked) to the specific asset being analyzed in the SLE calculation. It also coaxes risk analysts to think in broader terms and to look at the bigger picture when considering the risks associated with the compromise of a given asset. Thus, the introduction of CTM will help in the analytical discussion and an accurate valuation and calculation of a meaningful ROI. This lends credence to the efficacy of the selected approach used to determine the effectiveness of deploying IDS technology into a given network.

With the introduction of the cascading threat multiplier (CTM)—a multiplying factor—the definition of Single Loss Expectancy (SLE) is expanded. CTM, although somewhat subjective, is introduced mainly for the purpose of adding "flavor" to SLE. The formula for CTM is as follows:

Cascading Threat Multiplier (CTM) = 1 + ((UEA x EFS) / AV)

In this formula, Underlying Exposed Assets (UEA) is measured in dollars. These are the assets that are now exposed due to the compromise of a specific asset. Asset Value (AV) is identical to the calculation described elsewhere in this paper. Exposure Factor (EFS) represents Secondary Exposure Factor and is related to the percentage loss on the UEAs. Secondary Exposure Factor (EFS) is very similar to Exposure Factor (EF), as described in the standard equation in Table 2, with a few minute differences.

The primary reason for introducing EFS is to factor in the importance of an asset's logical location within a network. For example, if the asset is a Web server that is in a true demilitarized zone (DMZ) and has no access into the network or to any other corporate servers, EFS would be low since it is unlikely that an attacker can use this device to further compromise the network. But if the asset is on the same broadcast domain as other servers are on (such as e-mail, DNS and FTP), or there is no access control between the asset and other servers, then EFS will be higher. Finally, if the asset is on a network that has access to the rest of the network, then EFS will be very high. Examples of this would include hosts that offer some public services but are terminated within the internal network or hosts that have valid SSH keys to all other hosts.

It is important to consider what assets are easily (or even not so easily) accessible from a specific networked asset once that asset is compromised. When a given asset is compromised and used as a staging

Table 2 **ROI Variables and Risk Equations**

Variable	New Concept/ Expression	Formula or Expression
AV (Asset Value)		AV = hardware + comm. software + proprietary software + data
EF (Exposure Factor)		EF is the % estimation of the exposure of the initial compromised asset
UEA (Underlying Exposed Assets)		UEA is the estimation of the $ value of the assets behind the compromised initial asset
EFS (Secondary Exposure Factor)		EFS is the % estimation of the exposure of the UEAs
CTM (Cascading Threat Multiplier)	New	CTM = 1 + ((UEA x EFs) / AV)
SLE (Single Loss Expectancy)	New	SLE = EF x AV x CTM
ARO (Annual Rate of Occurrence)	New	ARO is estimated number, based on available industry statistics or experience
ALE1 (Annual Loss Expectancy without IDS)	New	ALE1 = SLE x ARO
ALE2 (Annual Loss Expectancy with IDS using auto-response)	New	ALE2 = conservative 50% reduction of ARO when IDS is managed skillfully with auto-response
ALE3 (Annual Loss Expectancy with IDS using auto-response and incident response)	New	ALE3 = conservative 25% reduction of EF and EFS when IDS is managed skillfully with auto-response and incident response
T (Annual Cost) of IDS Technology and Mgmt		T
R (Annual Recovery Cost) from Intrusions without IDS)		R = ALE1
E (Annual Dollar Savings) gained bystopping intrusions with IDS	New	E = ALE1 - (ALE2 or ALE3)
ROSI (Traditional Return on Security Investment) equation		ROSI = R - ALE, where ALE = (R - E) + T
ROI1 (GAIC ROI of IDS with auto-response)	New	ROI1 = ALE1 - ((ALE1 - (ALE1 - ALE2)) + T)
ROI2 (GAIC ROI of IDS with auto-response and incident response)	New	ROI2 = ALE1 - ((ALE1 - (ALE1 - ALE3)) + T)

point for attacks on other assets inside and outside a company's network, it could have potentially devastating consequences for the organization. If an attack is staged from the compromised asset to another asset outside the organization, even if the owner was not directly involved in the malicious activity, they can and probably will be held accountable. One can envision the UEA factor of SLE representing some portion of a trusted business partner's assets. It is easy to imagine the negative business impact the offending organization would encounter if one of their compromised assets were used as a staging ground to compromise and damage their business partner's assets.

What is the risk, quantified in dollars, of not considering a business partner's assets when performing a valuation exercise on your company's assets, ones which, if compromised, may enable access to more sensitive data and systems? The CTM concept provides the analytical framework to closely scrutinize the assets under an organization's control, assign more comprehensive valuations to those assets, and to more accurately measure the impact that a compromise of these assets could have on the organization.

As a practical example, we assume that a Web server has been compromised and used by a malicious person to stage attacks on other networked assets containing critical data valued at 10 times the amount (in dollars) of the data contained on the compromised Web server.

As the perpetrator hopscotches his way from asset to asset, penetrating deeper and deeper into the network, he may finally gain access to critical data on a vulnerable asset deep inside the company's network. The CTM for the Web server would be calculated as follows if we surmise (best estimate or WAG) that the Secondary Exposure Factor of the Underly-

ing Exposed Asset(s) is 70 percent:

$$CTM = 1+((10 * .7) /1) = 8$$

Thus, the CTM has increased the SLE for the compromised Web server by a factor of 8. We can follow the white arrow originating from the compromised Web server to better visualize this concept in Figure 3.

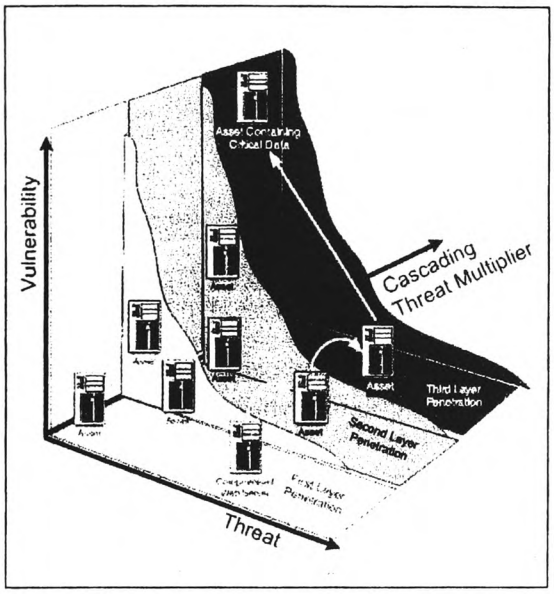


Figure 3 **Cascading threat multiplier**

Tying the CTM concept back into the SLE calculation, a new definition of Single Loss Expectancy can be expressed as:

$$SLE = EF \times AV \times CTM$$

A thorough risk management exercise should factor in the CTM concept by executing a more comprehensive valuation methodology, which included more subjective, intangible factors into their Asset Value (AV) variable calculation. As mentioned above, goodwill (i.e., business and consumer loyalty built on trust) and opportunity costs (i.e., choosing not to consider the effect that a compromised asset can have on other assets) are somewhat analogous to the CTM concept when these intangibles are factored into the Asset Valuation (AV) used in the SLE calculation.

The importance of capturing intangible value, and understanding the risks associated with jeopardizing the value, is one of the more challenging aspects of risk and return analysis. By introducing the CTM concept into the traditional SLE calculation we are attempting to make the capture of the intangible aspects of asset valuation a little less daunting of a task.

The risk analysis calculations listed above can be tied to an accepted formula for calculating the ROI for a security product.

$$\text{Return on Investment (ROI)} = \text{Recovery Cost (R)} \,.\, ALE$$

Where ALE = (R-E) + T, and E equals the savings gained by preventing an attack and T equals the cost of a security product. In Table 2 (previous page), each variable and risk equation is itemized for risk analysis and ROI calculations for IDS deployment in complex infrastructure. A review of the table shows how the traditional Return on Security Investment ROSI equation has been tied back to the ALE containing the CTM factor, i.e.,

$$ROSI = R - ALE,$$

Where the commonly accepted ALE = (R - E) + T is now replaced with

$$ALE = ARO * SLE,$$

And

$$SLE = AV * EF * CTM$$

Consequently, when all the numbers are tied together, the efficacy of the devised methodology for security product (e.g., IDS) risk assessment and ROI calculation can be demonstrated through a case study approach.

In the final analysis, conducting an effective risk assessment in complex infrastructures is entirely dependent upon a good valuation of networked security devices like the IDS.

7.0 Conclusion

There are several approaches to consider when conducting risk assessments for networked security products like Intrusion Detection Systems (IDS). During a risk assessment, the assessment team utilizes a multi-dimensional approach in anticipating on capitalizing on the benefits of both the qualitative and quantitative assessment approaches. Both assessments methodologies have unique benefits, as well as their own weaknesses.

The goal of any risk assessment is to ensure that the security of the computer systems is cost effective, up-to-date, and that the countermeasures in place are responsive to the various threats. There is

no one technique that can be relied upon as being the best. The product of the method selected will be as good as the input.

As noted earlier, it has been extremely difficult, if not impossible, for security products in complex environments to be assessed properly because of the difficulty to come up with asset values or replacement costs within the organization—elements critical to a risk analysis. SLE and ALE calculations are highly dependent on asset values in order to derive ROI of various mitigation strategies and safeguards. If accurate asset replacement values cannot be obtained, the quantitative analysis has no value. You simply cannot quantify something you can't obtain figures for.

Several researchers have pointed out the importance of using IDS for risk management. To effectively conduct a risk analysis of IDS, there is the need to have a sound understanding of the company, including at a minimum, the business practice, network topology and asset values. Equally, a good analysis of system vulnerabilities and associated threats should be addressed within the framework of a sound security policy and risk mitigation techniques.

Therefore, we can conclude that without quantifiable asset information, a quantitative analysis would provide little relative value to the organization. Finally, a positive risk assessment of security products like the IDS is attainable with an effective technique that utilizes the concept of cascading threat multipliers. ■

***Charles Iheagwara**, P.E., CCNP, GAIC, MCSE, is IT Security Director for Aligned Strategies Development, Inc. (2000 L Street, N.W., Suite 200, Washington, D.C., 20036). He is also Adjunct Professor of Computer Science at Strayer University (15 & L Street, N.W., Washington, D.C., 20036).*

References

[BRE99] *Brewer, Dr. David.* "Risk, Security and Trust in the Open World of E-Commerce." May 1999. URL: http://www.itsecurity.com/papers/p35.htm (22 March 2002).

[BRE00] *Brewer, Dr. David.* "Risk Assessment Models and Evolving Approaches." IAAC workshop, London. July 2000. URL: http://www.gammassl.co.uk/topics/IAAC.htm (22 March 2002).

[CAI01] "CAIDA Analysis of Code-Red." 15 August 2001. URL: http://www.caida.org/analysis/security/code-red/ (22 March 2002).

[CCC02] "Overview of Attack Trends." 19 February 2002. URL: http://www.isalliance.org/resources/papers/attack_trends.pdf (22 March 2002).

[CHA99] *Changduk, J., Han, I., Bomil, S.* "Risk Analysis for Electronic Commerce Using Case-Based Reasoning." 1999. URL: http://afis.kaist.ac.kr/download/inter_jnl012.pdf (22 March 2002).

[CHI97] *Chisnall, W. R.* "Applying Risk Analysis Methods to University Systems." EUNIS 97, European Cooperation in Higher Education Information Systems, Grenoble, France. 9-11 September 1997. URL: http://www.lmcp.jussieu.fr/eunis/html3/congres/EUNIS97/papers/022701.html (22 March 2002).

[COM02] "Computer Economics Security Review 2002." URL: http://www.computereconomics.com/cei/news/secure02.html (22 March 2002).

[CRA98] *Craft, R., Wyss, G., Vandewart, R., Funkhouser, D.* "An Open Framework for Risk Management." 21st National Information Systems Security Conference Proceedings. October 1998. URL: http://csrc.nist.gov/nissc/1998/proceedings/ paperE6.pdf (22 March 2002).

[CSI01] "Financial losses due to Internet intrusions, trade secret theft and other cybercrimes soar." March 2001. URL: http://www.gocsi.com/prelea/000321.html (22 March 2002).

[CST02] "CERT/CC Statistics 1988-2001." URL: http://www.cert.org/stats/cert_stats.html (22 March 2002).

[CUG02] "About CRAMM." URL: http://www.crammusergroup.org.uk/cramm.htm (22 March 2002).

[CUS01] CRAMM User Guide, Issue 2.0. Walton-on-Thames: Insight Consulting, January 2001.

[DOS01] "New Private-Sector Internet Security Alliance Launched." 23 April 2001. URL:http://usinfo.state.gov/topical/global/ecom/01042303.htm (22 March 2002).

[GAM97] "A Practitioner's View of CRAMM." September 1997. URL: http://www.gammassl.co.uk/topics/hot5.html (22 March 2002).

[GIL89] *Gilbert, I.E.* "Guide for Selecting Risk

Analysis Tools." NIST Special Publication 500-174. October 1989. URL: http://csrc.nist.gov/publications/nistpubs/500-174/sp174.txt (22 March 2002).

[KEV02] *Kevin, T., Kinn, D., CTM*. Technical Report, Netsolve, Inc., Austin, USA 2002

[KRA99] *Krause, M., Tipton, H.F.,* "Section 3-1: Risk Analysis." Handbook of Information Security Management. December 1999. URL: http://secinf.net/info/misc/handbook/242-244.html (22 March 2002).

[LAB99] *Labuschagne, L., Eloff, J.H.P,* "Risk Analysis Generations—The Evolution of Risk Analysis." August 1999. URL: http://csweb.rau.ac.za/deth/research/articles/ra_generations.pdf (22 March 2002).

[Harris] *Harris, S.* CISSP Certification Guide. McGraw-Hill/Osborne, 2001, pp 72-91.

[NIS91] "Description of Automated Risk Management Packages that NIST/NCSC Risk Management Research Laboratory have examined." March 1991. URL: http://www.eff.org/Privacy/Newin/New_nist/risktool.txt (22 March 2002).

[OZI99] *Ozier, W.* "A Framework for an Automated Risk Assessment Tool." 15 August 1999. URL:http://www.theiia.org/itaudit/index.cfm?fuseaction=forum&fid=228 (22 March 2002).

[SCO01] *Hinton, C.* "CRAMM." December 2001. URL: http://www.scmagazine.com/scmagazine/sc-online/2001/review/059/product.html (22 March 2002).

[USG01] United States General Accounting Office, "Information Security: Weak Controls Place DC Highway Trust Funds and Other Data at Risk" in *Report to the Mayor of the District of Columbia*, (GAO-01-155), January 2001.

[VEN99] *Venter, H.S., Labuschagne, L., Eloff, J.H.P.* "Real-time Risk Analysis on the Internet." March 1999. URL: http://csweb.rau.ac.za/ifip/workgroup/docs1999/11_sec1999.doc (22 March 2002).

[RISKIT] http://dacs.dtic.mil/awareness/newsletters/technews2-2/riskit-references.html

Appendix 7

"The Impact of IDS Deployment Technique on Threat Mitigation" In: Proceeding of the International Conference on Industrial Engineering and Engineering Management (IE&EM'2003), Shanghai, China, on December 6-8 2003.

Proceedings of IE&EM'2003
2003 International Conference of Industrial Engineering and Engineering Management:
Global Industrial Engineering in e-century
Dec. 6-8, 2003. Shanghai, P. R. China

RID: D033

THE IMPACT OF IDS DEPLYMENT TECHNIQUE ON THREAT MITIGATION

Charles Iheagwara [1] Andrew Blyth [2]

[1.] Aligned Development Strategies, Inc., 1925 K Street, NW, Suite G2, Washington, DC 20006, USA Email: iheagwarac@aol.com

[2.] School of Computing, University of Glamorgan, Pontypridd, CF 37 1DL, Wales, UK
Email: aicblyth@glam.ac.uk

ABSTRACT

In this paper we explore IDS deployment techniques and risk analysis methodologies. We discuss general IDS technologies and expand on the impact that the logical location of a company's critical networked assets could have on the risk equations. To this end we introduce the Cascading Threat Multiplier (CTM) to expand on the Single Loss Expectancy

(SLE) equation. We also review commonly accepted risk equations. We examine the effect of IDS management techniques on the annual loss expectancy. We propose new formulas for accurate risk analysis valuations culminating in a new formula for calculating ROI for security, otherwise commonly known as Return on Security Investment (ROSI). Finally, we demonstrate the efficacy of this equation through a casestudy.

Keywords: Intrusion Detection, Risk Assessment.

1 Introduction

The recent CSI-FBI survey [1] of 503 American organizations validated the continued concerns of business leaders today with doing business in the electronic era. Of the 503 organizations surveyed, 90% detected a security breach of their information systems and 80% experienced financial losses as a result of breaches. While internal threats remain a top priority, 40% cited breaches from outside their organization. Additionally, 85% experienced viruses and 74% stated their Internet connection was most frequently targeted. The most signification piece of data from this survey indicates that 90% of these respondents have a Web site, 90% have firewalls and antivirus programs and 100% conduct business electronically in some fashion.

The statistics in the survey points to a notable trend, not necessarily the percentages, but simply that 100% of those surveyed are conducting business electronically and 90% of them have firewalls and antivirus, yet 90% reported system breaches. Protecting information systems today must be done in a layered process, which includes technology and human analysis. As the CSI-FBI survey revealed, most companies have already deployed firewalls and antivirus programs, and many are moving aggressively towards acquiring Intrusion Detection Systems (IDS), a security system that monitors computer

systems and network traffic and analyzes that traffic for possible hostile attacks originating from outside the organization and also for system misuse or attacks originating from inside the organization.

But given the high cost of an IDS deployment especially when multiple deployments are involved, organizations must justify implementation expenses by proving that the IDS is a value added resource. The justification is to prove that the deployment of the IDS is going to lead to a reduction in the annual loss expectancy (ALE) and the return on security investment (ROSI). This is realized if the IDS is able to effectively detect and deter attacks.

One method for justifying IDS is by determining the value of the ALE using conventional cost/benefit (risk) assessment; the

ALE represents the cost/benefit break-even point for risk mitigation measures. In other words, the organization could justify spending up to the dollar amount equivalent of the ALE per year to prevent the occurrence or reduce the impact of a fire. A risk assessment can identify what types of intrusions a company's infrastructure is vulnerable to and the potential for loss should an attack occur. It will also provide the justification of IDS deployment as an effective safeguard. Another way to analyze the benefits of IDS is to document the misuses of an organization's network. The CSI-FBI survey shows that 78% of the respondents detected employee misuse of its systems and its internet connection. This included web surfing, email abuse, and use of company hardware/software for personal gain. This misuse directly increases the risk of systems being attacked and information compromised, which can be tied to justifying the need and expense of IDS.

An alternative method for justifying IDS is to demonstrate the ability of the IDS to effectively detect and deter attacks in quantifiable measures. There are performance studies [2,3,4] that demonstrate the different aspects to this. A more elaborate discussion on the performance studies is given in Section 2.

In this paper, we review risk analysis methodologies, introduce new (CTM) concepts into risk equations, explore the impact of threat mitigation on the annual loss expectancy (ALE) and correlate the effect of IDS management technique on threat mitigation.

The rest of the paper is organized as follows. In Section 2 we discuss current IDS deployment and implementation methods. We discuss risk assessment methodologies in Section 3. In Section 4 we present a novel concept to risk analysis – the Cascading Multiplier Effect (CTM) and discuss the relation of the annual loss expectancy (ALE) to threat mitigation in Section 5. We then present a case study to illustrate the impact of deployment techniques on ALE reduction and a positive Return On Investment (ROI) in Section 6.

1 IDS Technologies and Deployments in Complex Environment

Intrusion detection is an overlay of two separate and different (NIDS) and Host-based IDS (HIDS) technologies. The primary advantage of NIDS is that it can watch the whole network or any subsets of the network from one location. Therefore, NIDS can detect probes, scans, and malicious and anomalous activity across the whole network. These systems can also serve to identify general traffic patterns for a network as well as aid in troubleshooting network problems. When enlisting auto-response mechanisms, NIDS can protect independent hosts or the whole network from intruders. NIDS does, however, have several inherent weaknesses. These weaknesses are its susceptibility to generate false alarms, as well as its inability to detect certain attacks called false negatives. NIDS also is not able to understand host specific processes or protect from unauthorized physical access.

HIDS technology overcomes many of these problems. However, HIDS technology does not have the benefits of watching the whole network to identify patterns like NIDS does. A recommended combination of host and network intrusion detection systems, in which a NIDS is placed at the network border and an HIDS is deployed on critical servers such as databases, Web services and essential file servers, is the best way to significantly reduce risk.

Generally speaking, most of these host-based systems have common architectures, meaning that most host systems work as host agents reporting to a central console. The associated cost of HIDS deployments can vary depending on vendor and software versions. A good baseline is that agents can cost between $500 and $2000 each and consoles may cost in the $3000-$5000 range. This does not include OS, hardware or maintenance costs. Network intrusion detection systems can be deployed as stand-alone hosts with a possible management interface or distributed sensors and management console. Generally speaking, commercially available sensors run in the $5000-$20,000 area depending on vendor, bandwidth and functional capabilities. Management consoles can be free or can cost several thousand dollars depending on the vendor. This does not necessarily include hardware or back-end databases. The total cost of an IDS deployment depends on implementation costs combined with the costs for managing the technology.

In the enterprise, IDSs are implemented as either a single or multiple deployments. Multiple IDS deployments are intended to solve the problem of the high volume traffic stream that are increasingly becoming common place in today's enterprise network systems that represent a vast array of complex technologies often with highly switched topology. A big reason many networks operate in a switched environment can be attributed to the security/performance benefits [5].

In considering the implementation of any IDS technology, a return on investment can be understood by analyzing the difference between annual loss expectancy (ALE) without IDS deployment and the ALE with IDS deployment, adjusted for technology and management costs. The ultimate initial goal, then, should be to prove that the value proposition (re: a benefit in the form of a quantifiable reduction in ALE) in implementing and effectively managing the IDS technology is greater than the implementation and management costs associated to deploying the IDS technology.

A positive return on investment (ROI) of intrusion detection systems (IDS) is dependent upon an organization's deployment strategy and how well the successful implementation and management of the technology helps the organization achieve the tactical and strategic objectives it has established. For organizations interested in quantifying the IDS's value prior to deploying it, their investment decision will hinge on their ability to demonstrate a positive ROI. ROI has traditionally been difficult to quantify for network security devices, in part because it is difficult to calculate risk accurately due to the subjectivity involved with its quantification. Also, business-

relevant statistics regarding security incidents are not always available for consideration in analyzing risk.

As companies race to deploy more IDSs to meet the demands placed by Gigabit traffic the difficulty for accurate risk calculations will multiply. Another concern is how best to deploy the IDS so as to maximize performance benefits. This is a legitimate concern because of the correlation between the IDS mitigation ability and deployment technique as has been shown in Section 5. Studies [2] have demonstrated the effect of deployment techniques on the performance of the IDS. Equally, the benefits of multiple deployments or the use of Gigabit sensors have been demonstrated [3]. Iheagwara et al [3] examines the system benefits of using a single Gigabit IDS sensor instead of multiple Megabit sensors (see Figure 1 in Appendix 1) for a wide range of defined system attacks, network traffic characteristics, and for their contexts of operational concepts and deployment techniques. The experimental results were analyzed in the context of practical experiences in the operation of these IDS systems. The results of this analysis provide the probabilistic framework (Figure 2 in Appendix 2) for the performance measure of the IDS within a Gigabit traffic stream. Specifically, the study point to the fact that the IDS detection rates increase with the number of sensors deployed thereby making the case for multiple deployments.

There is, therefore, every reason to believe that multiple IDS deployments will increasingly become a common place as companies scale up to Gigabit bandwidth. As noted before, ROI has traditionally been difficult to quantify for network security devices, in part because it is difficult to calculate risk accurately due to the subjectivity involved with its quantification. The difficulty becomes even more arduous in environments with multiple deployments, highly networked and interdependent. Hence, devising an effective technique or methodology for accurate risk analysis of multiple IDSs assumes a great importance.

Devising effective risk analysis technique for the IDS in complexity environments requires a re-examination of the basic concepts, assessment approach, and risk analysis formulas. Pertinent questions to which answers are sought include: how do we accurately assess the value and hence the effectiveness of the IDS? And what deployment techniques will have the most positive impact on threat mitigation and reduce the annual expectancy loss of protected assets? Before we explore the answers to these questions in Sections 4.0 and 5.0 let us review in the next Section risk assessment methodologies and the challenges of IDS risk assessment in complex environments.

3 Assessment Methodologies

The landscape of risk assessment methodologies is constantly changing. Some methodologies promulgated via U.S. government FIPS 65 guideline for performing risk analysis in large data processing centers [6] were withdrawn in 1995. Recent approaches to risk assessment attempt to adapt technological advances like the Internet by prototyping real-time risk analysis [7] and emerging applications like e-commerce by using case-based reasoning [8] or considering a framework for the "whole system" during the risk management life cycle [9]. In the latter work and especially in [10], three generations of risk analysis and management methodologies are identified in which the first generation corresponds to the mainframe era, the second to networks and distributed computing, and the third to open environments and the Internet and their shortcomings discussed.

The following are the widely used methodologies in risk analysis.

3.1 Quantitative and Qualitative Methodologies

There are two broad approaches for risk assessment: quantitative and qualitative methodologies. In practice, none of the two are sufficient for a robust risk assessment of intrusion detection systems in complex environments – hence a new technique or assessment approach is necessary.

The quantitative approach articulates risk in numerical terms, i.e. expected monetary loss and probability (e.g. annual loss expectancy, ALE). The qualitative approach has no numeric value and is usually opinion based. Results are summarized in words like "low", "medium" and "high".

Quantitative and qualitative risk analyses offer two perspectives to the assessment of risk. The argument for justifying quantitative risk assessment is that cost-effective safeguards cannot be evaluated against losses unless the risks are quantified. This is not tenable with qualitative methodologies that emphasize descriptions rather than calculations. Quantitative risk assessments make use of a mathematical calculation produced from the probability of an event occurring and the likely loss should it occur to assign real numbers to the costs of countermeasures and the amount of damage that can take place [11]. This is called the Annual Loss Expectancy. Probability can rarely be precise and can, in some cases, promote complacency. In addition, controls and countermeasures often tackle a number of potential events and the events themselves are frequently interrelated. In qualitative risk assessment, probability data is not required and only estimated potential loss is used.

The quantitative approach also provides concrete probability percentages when determining the likelihood of threats and risks. Each element within the analysis (asset value, threat frequency, severity of vulnerability items) is quantified and entered into equations to determine total and residual risks. Purely quantitative risk analysis is not possible because the method is attempting to quantify qualitative items. If a severity level is high and a threat frequency is low, it is hard to assign corresponding numbers to these ratings and come up with a useful outcome.

The classic quantitative algorithm, as presented in FIPSPUB-65 [6] laid the foundation for information security risk assessment:

(Asset Value x Exposure Factor = Single Loss Expectancy) x

Annualized Rate of Occurrence = Annualized Loss Expectancy. (1)

For example, let's look at the risk of fire. Assume the Asset Value is $1M, the exposure factor is 50%, and the Annualized Rate of Occurrence is 1/10 (once in ten years). Plugging these values into the algorithm yields the following:

($1M x 50% = $500K) x 1/10 = $50K

Using conventional cost/benefit assessment, the $50K ALE represents the cost/benefit break-even point for risk mitigation measures. In other words, the organization could justify spending up to $50K per year to prevent the occurrence or reduce the impact of a fire.

The qualitative approach does not assign monetary values to components and/or losses. Instead, qualitative methods walk through different scenarios of risk possibilities and rank the seriousness of the threats and the sensitivity of the assets. Qualitative analysis techniques include judgment, intuition, and experience. Examples of qualitative techniques are Delphi, brainstorming, storyboarding, focus groups, surveys, questionnaires (Table 1 in Appendix 3), checklists, one-on-one meetings, and interviews.

3.2 The Challenges of IDS Risk Assessment in Complex Environments

The idea/concept of an IDS risk assessment is to demonstrate through the chosen methodology that the organization will suffer immensely if the IDS is not available in the event of an intrusion. Answers gathered from a formal risk assessment can help establish companies' valid business reasons for adding IDS to their infrastructure. The risk formulas used in the current methodologies do not factor in the new concepts presented in Section 4, which are needed to integrate the new elements introduced by technological improvements and changing landscapes.

Generally speaking, the risk assessment methodology for IDS follows the same methodology that is used in other assessments. However, a major difference is that performing an IDS risk assessment is like trying to determine the return on investment. In addition, because there are different deployment configurations i.e. deployment of multiple IDS sensors in the combination of firewalls, filtering routers, etc. the risk assessment effort in itself becomes determining how much the IDS contributes to the defense of the network.

In today's complex environments with increasingly large number of complex network architectures, the challenges become more profound. The enterprise network as a system which interconnects a multitude of computers and devices for the purpose of communications and information/resource sharing are complex environments trying to balance policy priorities, user expectations, technological development and demands, and scalability issues while under changing economic constraints. Developments in technology overwhelm almost every factor in its balance.

The complexity of the network environments gets more compounded with the addition of new technological resources used to integrate and centralize the enterprise systems, in order to control access to protected data. The technology of computer networks, on the other hand, promotes a mode of work that goes against all centralized efforts. Also with the growth of networks comes the migration of applications from centralized systems to client-server environments. In addition, organizations are connecting their networks to those of other organizations and to the Internet at a rapid rate. All of this added complexity presents a challenge to risk assessors who are responsible for making sure that the basic elements of risk assessment in such environments are accurate and takes into account all of the above mentioned relational schemes.

Also, the practical reality is that the tasks involved in risk assessments can be overwhelming. This is because the process tends to establish metrics as yardsticks to proffer remedies while at the same time try to maintain to a considerable degree a high level of accuracy. In complex environments, conducting risk assessments become even more complex as a result of the difficulties created by the interplay of people, technology and operations. In this case, risk determination even with the simplest asset could be complex.

The determination of the asset value when there is interdependence in networked environments could be extremely difficult. This is because the asset value can be taken in up and down stream dimensions. An asset value can be measured in so many dimensions and in tangible and intangible measures. This challenge is encountered in complex infrastructures where the valuation of mission critical security devices like intrusion detection products is difficult.

It is therefore, obvious that the current quantitative risk formulas are devoid of the new concepts that provide the analytical framework for accurate valuation of IDS devices that are deeply enmeshed in a complex web of new technological environments. The fact is that the technological environments are constantly changing while the risk formulas crucial for accurate valuation have not changed in commensurate proportions. The importance of a good risk assessment cannot be over stated because only the accurate valuation will lead to the establishment of the true cost of the IDS, which will be used to judge its performance against the cost of the asset that it is trying to protect.

In the final analysis, conducting an effective risk assessment in complex infrastructures is entirely dependent upon a good understanding of the environment and the accurate valuation of networked security devices like the IDS. In Section 4.0, we introduce the new concepts that tie the intangible factors into risk formulas and present through our analytical discussion a modular approach for the assessment of IDS devices in complex infrastructures.

4 Novel Concepts: The cascading Threat Multiplier

The interplay of technological processes, policies and risk management methods in today's enterprise environments requires the formulation of new analytical frameworks and concepts like the Cascading Threat Multiplier (CTM) to accurately conduct valuation studies and quantify the return on investment (ROI) for any acquired or developed technology.

The CTM factors in the importance of other critical assets tied (re: networked) to the specific asset being analyzed in the SLE calculation. It also coaxes risk analysts to think in broader terms and to look at the bigger picture when considering the risks associated with the compromise of a given asset. Thus, the introduction of the Cascading Threat Multiplier (CTM) will help in the analytical discussion and an accurate valuation and calculation of a meaningful ROI. This lends credence to the efficacy of the selected approach used to determine the effectiveness of deploying IDS technology into a given network.

With the introduction of the Ca scading Threat Multiplier (CTM) - a multiplying factor, the definition of Single Loss Expectancy (SLE) is expanded. CTM, although somewhat subjective is introduced mainly for the purpose of adding, "flavor" to SLE.

In our analytical approach working up to the calculation for ROI, we will use commonly accepted formulas and definitions associated with asset valuation, exposure, threat, vulnerability and loss expectancy. The Cascading Threat Multiplier (CTM), an additional factor we've added to the mix, enables us to expand on the widely accepted calculation for Single Loss Expectancy (SLE) where, traditionally, SLE = Exposure Factor (EF) x Asset Value (AV).

In order to stress the importance of the intangible considerations that will help us apply our holistic approach for quantifying risk and calculating a meaningful ROI, the concepts of goodwill and opportunity costs should be considered when performing valuation exercises on company assets. Although intangible factors inherently introduce subjectivity into risk and return analysis, it is nonetheless an important step to consider intangibles before one can arrive at a more meaningful calculation of ROI. It is worth mentioning here that, in general, it may be safe to assume that organizations would tend to undervalue certain data assets if they have not fully taken into account (or bothered to understand for that matter) how these assets relate to the "big picture". It is simple human nature to take the path of least resistance when given a choice. But that's a very dangerous path to take for anyone attempting to arrive at an accurate assessment of the value of data assets residing on their network. Understanding the tangible costs and benefits of an asset is much easier than understanding, or even considering for that matter, the intangible costs and benefits associated to that same asset. Clarifying this understanding is one of our challenges and one we will address throughout the rest of the article as we work toward calculating the IDS ROI for Wally's Building Supplies, Inc.

The following are the commonly accepted risk/return analysis definitions and formulas[11]:

> ***Asset value:*** One can measure an informational assets value by estimating the development, purchasing, licensing, supporting and replacement costs associated with the resource. Value can also be measured from an organizational as well as an external market perspective. The asset value is represented as follows:

Asset Value (AV) = hardware + comm. software + proprietary software + data. (2)

> ***Exposure Factor:*** The Exposure Factor (EF) represents the percentage of loss that a realized threat could have on a specific asset when the specific threat matches up with a specific vulnerability. A threat is a single event that has the potential to cause damage to an asset and vulnerability is a known or unknown weakness that can be exploited by any number of known or unknown threats. The threat usually manifests itself through vulnerability in the information system.

> ***Single Loss Expectancy:*** In the end, risk is evaluated in terms of money. This is true even if life is lost; in the case of loss of life, it may be a lot of money. For any threat we have defined, we take the value of assets at risk and multiply that by how exposed they are. This yields the expected loss if we were to get clobbered by the threat. This is called the single loss expectancy (SLE) and is expressed as

(SLE) = EF x AV. (3)

> ***Annual Loss Expectancy:*** The Annual Loss Expectancy (ALE) is the annually expected financial loss to an asset resulting from one [specific] threat. The Annual Rate of Occurrence (ARO) is the estimated number of times a threat on a single asset is estimated to occur. The higher the risk associated to the threat the higher the Annual Rate of Occurrence. The expression is given as

ALE = SLE x ARO. (4)

Now let's introduce a new concept, Cascading Threat Multiplier (CTM), into the mix. This will greatly aid us in our analytical discussion and move us further along in distilling a meaningful ROI calculation that can help us determine the effectiveness or ineffectiveness of deploying IDS technology into a given network.

The Cascading Threat Multiplier (CTM) shown in Figure 3 (Appendix 4) is a multiplying factor that will be included into our expanded definition of Single Loss Expectancy (SLE). CTM is somewhat subjective and is introduced mainly for the purpose of adding a little more "flavor" to SLE. CTM factors in the importance of other critical assets tied (re: networked) to the specific asset being analyzed in the SLE calculation. It also

coaxes us to think in broader terms and look at the bigger picture when considering the risks associated to the compromise of a given asset. The formula for CTM is as follows:

(CTM) = 1 + ((UEA x EFS) / AV). (5)

In this formula, Underlying Exposed Assets (UEA) is measured in dollars. These are the assets that are now exposed due to the compromise of a specific asset. Asset Value (AV) is identical to the calculation previously described above. Exposure Factor (EFS) represents secondary exposure factor and is related to the percentage loss on the UEAs. Secondary Exposure Factor (EFS) is very similar to Exposure Factor (EF), as previously described in the standard equation above, with a few minute differences. The primary reason for introducing EFS is to factor in the importance of an assets logical location within a network. For example, if the asset is a Web server that is in a true DMZ and has no access into the network or to any other corporate servers, EFS would be low since it is unlikely that an attacker can use this device to further compromise the network. But if the asset is on the same broadcast domain as other servers are on (such as e-mail, DNS and FTP), or there is no access control between the asset and other servers, then EFS will be higher. Finally, if the asset is on a network that has access to the rest of the network, than Secondary Exposure Factor (EFS) will be very high. Examples of this would include hosts that offer some public services but are terminated within the internal network or hosts that have valid SSH keys to all other hosts.

It is important to consider what assets are easily (or even not so easily) accessible from a specific networked asset once that asset is compromised. When a given asset is compromised and used as a staging point for attacks on other assets inside and outside a company's network, it could have potentially devastating consequences for the organization. If an attack is staged from the compromised asset to another asset outside the organization, even when the owner was not directly involved in the malicious activity, they can and probably will be held accountable. One can envision the UEA factor of SLE representing some portion of a trusted business partner's assets. It is easy to imagine the negative business impact the offending organization would encounter if one of their compromised assets were used as a staging ground to compromise and damage their business partner's assets. What is the risk, quantified in dollars, of not considering a business partner's assets when performing a valuation exercise on your company's assets, ones that, if compromised, may enable access to more sensitive data and systems? The CTM concept reminds us to closely scrutinize the assets under our control, assign more comprehensive valuations to those assets, and more accurately try to measure the impact that their compromise could have on the organization.

Let's assume that a Web server is compromised and used by a malicious person to stage attacks on other networked assets containing critical data valued at 10 times the amount (in dollars) of the data contained on the compromised Web server. As the perpetrator hopscotches his way from asset to asset, penetrating deeper and deeper into the network, he may finally gain access to critical data on a vulnerable asset deep inside the company's network. The CTM for the Web server would be calculated as follows if we surmise (re: best guess or WAG) that the secondary exposure factor (EFS) of the Underlying Exposed Asset(s) (UEA) is 70%: CTM = 1+((10 * .7) /1) = 8. The CTM has increased the SLE for the compromised Web server by a factor of 8. Follow the white arrow originating from the compromised Web server to better visualize this concept.

Tying the CTM concept back into our SLE calculation, our new definition of Single Loss Expectancy is as follows:

SLE = EF x AV x CTM. (6)

Factoring in our CTM concept makes for a more comprehensive valuation methodology that includes intangible factors into their Asset Value (AV) variable calculation. As mentioned above, goodwill (i.e. business and consumer loyalty built on trust) and opportunity costs (i.e. choosing not to consider the effect that a compromised asset can have on other assets) are somewhat analogous to our CTM concept when these intangibles are factored into the Asset Valuation (AV) used in the SLE calculation. The importance of capturing intangible value, and understanding the risks associated to jeopardizing that value, is one of the more challenging aspects of risk and return analysis. By introducing our CTM concept into the traditional SLE calculation we are attempting to make the capture of the intangible aspects of asset valuation a little less daunting of a task.

All of the above concepts of risk management and other related equations are given in Table 2 (Appendix 5) will lead us to accurately calculate the annual loss expectancy (ARO).

In Section 5.0 we explore the different deployment techniques and analyze within the context of operational performance how each technique affect the annual loss expectancy (ARO).

5 Deployment Technique vs. Threat Mitigation

The measure of risk can be determined as a product of threat, vulnerability and asset values:

Risk = Asset x Threat x Vulnerability. (7)

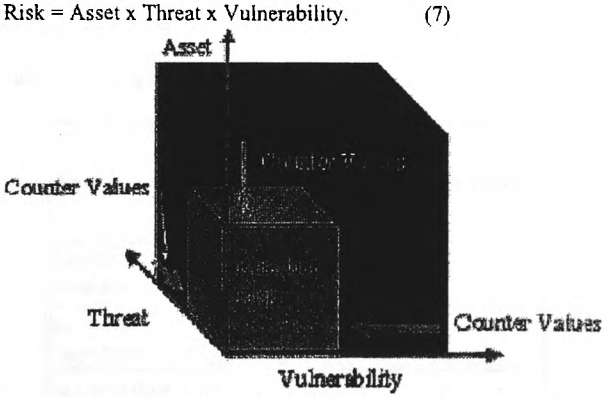


Fig. 4. Risk as a function of asset value, threat and vulnerability [12].

The risk elements and their corresponding countermeasures for a specified system can best be visualized with a cuboid (Figure 4). The system has an initial level of risk before any countermeasures are applied. Countermeasures, assuming that their values are assigned by the same parameters that are used for threat, vulnerability and asset valuation, can reduce risk, i.e. by reducing threat (e.g. locked doors, IDSs), reducing vulnerability (e.g. awareness, patches, hot fixes) or reducing asset value (e.g. encryption). After calculating the results from each combination of threat, vulnerability, asset and countermeasure the residual risk is determined [13]. Here the impact element is covered in asset value, the likelihood in threat and vulnerability values.

The effectiveness of the IDS as a countermeasure to reduce threat in an organization is very dependent on the deployment technique. Independent of implementation and management costs, the method in which the devices are deployed can have a serious effect on the annual loss expectancy and the return on investment (ROI). Two deployment and management techniques – proactive and reactive are generally implemented. To this point, the key question to answer is: is the system going to be proactive or reactive as security events are detected? The following in Table 3 depicts the normal event flow in each method. A proactive implementation response is automated by the system while a reactive implementation response is done once personnel have been enlisted.

Table 3. IDS Deployment schemes

Method	System actions	Personnel actions	Follow up information
Reactive	Log -> Alert ->	Respond -> Analyze ->Eradicate	Forensics and Evidence
Proactive	Respond -> Log -> Alert	Analyze -> Eradicate if necessary	Forensics and Evidence

By examining the Annual Loss Expectancy (ALE = ARO * SLE, where SLE = Exposure Factor * Asset Value * Cascading Threat Multiplier) we can determine which variables are affected by each of these two management methods. In a reactive design, where personnel must be engaged to respond to each event, the exposure factors (primary [EF] and secondary [EFS]) will be affected. In a proactive design there will be similar benefits to the exposure factors (re: a reduction) and, in addition, the Annual Rate of Occurrence (ARO) will be influenced in a beneficial way as well. To demonstrate the impact of threat vs. time (Figure 5 in Appendix 6) we will use the concept of primary and secondary mitigation windows. In the following graph the primary mitigation window affects ARO while the secondary mitigation window affects Exposure Factor and Cascading Threat Multiplier. An effective way of impacting ARO is through automated response.

Auto-response can take many forms. On host-based IDS this is sometimes called shielding, where a specific process is terminated. Network-based IDS generally employs TCP resets or shunning. TCP resets effectively kill one specific session based on suspicious activity, but it still allows other activity from that same IP. Shunning, on the other hand, changes firewall rules or router access lists and effectively denies all traffic from that host for a specific period of time. In essence, shielding will protect a single host from one process, resets will protect a host from a specific session, and shunning will protect the entire network from a specific host for a pre-determined amount of time.

The accuracy of automated response can vary tremendously. This is dependent on the skill level of the engineers managing the devices. If the engineers are moderately skilled then auto-response will not be very effective, which may adversely affect the ROI of the IDS deployment. This adverse effect may manifest itself in the form of a loss of productivity from network-related problems due to improperly implemented auto-response, as well as the additional fallout related to a false sense of security throughout the company.

With skilled engineers managing the devices, auto-response can be very accurate and effective. Because few statistics exist that illustrate the accuracy of automated response we will use statistics [14] generated from our analysis of one month's worth of data on networks that NetSolve, Incorporated manages. If we include Code Red and Nimda activity, in 99.96% of the attacks, where automated response was used to mitigate the threat, the activity was malicious. Excluding large-scale worms, the attacks were malicious in 95.8% of auto-response uses. Of the 4.2 % of the traffic that was not malicious, not all of it was desirable. Some of this traffic was peer-to-peer programs, on-line gaming, chat and other undesirable traffic that triggered alarms. The percentage of traffic that was denied that was business related was very small. It should be noted that many of these devices provide numerous different techniques for ensuring that very little, if any, legitimate traffic is denied through the use of automated response.

Table 4. Average Attack Occurrence per Network

Attack	Per Network Attempts (April)	Per Year Attack Attempts	Scenario
General Cmd.Exe	9492	113,904	1
Root.exe backdoor	1869	22,428	1
Ida overflow	105	1260	1
SSH Attacks	.2	2.0	2

DNS Bind Attacks	.7	8.0	2
FTP Attacks	.3	4.0	2
Apache Chunked	.6	7.0	2
IOS HTTP Unauth	3	36	3

To determine how effective the device is in recognizing attacks we will use the most recent NSS study [15]. In this test the worst NIDS detected 67 of 109 attacks or 61.5%, while the best detected 94 of 109 attacks for an 86.2 % detection rate. Even the worst case, the 61.5% detection rate was out of the box and it was reported that it would not be difficult to improve this with some custom signatures and tuning?

What does all this mean? It means that the worst IDS tested can still detect at least 61.5% of attacks. Realistically that number should be closer to 70% when a skilled engineer or technician manages the device. This ultimately means that the auto-response feature, when properly used, can be a very effective method of intrusion detection (and hence avoidance) which ultimately reduces the Annual Rate of Occurrence (ARO).

In section 6.0, we present a case study to demonstrate how each deployment technique affect the annual loss expectancy (ALE) and the return on investment (ROI).

6 Case Study

This case study will permit the in-depth exploration of the benefits of performing risk analysis to maximize the management techniques of intrusion detection systems. From these, we hope to glean some general concepts about intrusion detection system ARO & ROI and determine the viability of the management approach that will enhance the maximization. By developing the examples, we also hope to develop a possible method of reasoning about IDS risk analysis more generally. The case study will be presented in the context of events and risk analysis in a hypothetical company called WBS, Inc.

6.1 Methodology

In Section 6.2 we describe the enterprise business setting of the company (Wally's Building Supplies) that we use in this case study. We then discuss the threats and attacks that compromised the security of the business in Section 6.3. Next, in Section 6.4 we analyze the attacks, compromises and contributing factors and delineate the sources of the security breach. Part of the analysis is the recommendation of the necessary safeguards to forestall future attacks and in this case deployment of intrusion detection systems. Based on the analysis results, we then calculate the annual loss expectancy ALE in Section 6.5. Finally, we summarize the results in Section 6.6.

6.2 The Wally's Building Supplies

The case study is a risk analysis of Wally's Building Supplies (WBS) shown in Figure 6 (Appendix 7), which experienced a VPN attack that resulted to a compromise of the company's assets. WBS has six supply outlets, with the business office located within the primary outlet. WBS has several business-to-business (B2B) VPN connections to its suppliers. Their small staging department procures most of WBS items for all six outlets over these B2B connections by running an over-the-counter order procurement software application agreed on by each of the suppliers. Of the dozen or so suppliers, ACME is WBS most important one, accounting for 50% of all WBS procurement needs. ACME and WBS have built their trust relationship over the course of many years doing business together. ACME has experienced phenomenal growth over the past decade and supplies scores of building suppliers around the country. WBS orders account for a mere 1% of total ACME sales.

For several years WBS has maintained a simple informational Web page showing store locations and directions, general goods and services available and monthly specials. The primary target market for WBS consists of residential and commercial building contractors. Contractors comprise 75% of total WBS sales, with the remaining 25% generated from do-it-yourself consumers.

Recently WBS had contracted out the development of a dynamic database-driven Web site that allows contractors to order supplies on-line, check the status of their orders, and confirm deliveries to the construction site. The dynamic Web site has already had a positive effect on the operational ROI of WBS by improving efficiencies related to its' antiquated order fulfillment and delivery confirmation process. Inventory turnover has increased as a result of these efficiency gains, which in turn has improved WBS bottom line. That's the good news. WBS maintains Internet connectivity through a T1. Most of WBS servers, routers and infrastructure have been set up by outside IT contractors. So what's the bad news?

6.3 Compromise

WBS primary supplier, ACME, recently informed WBS that a malicious attacker gained access to ACME's data and network through the VPN tunnel with WBS. It is unknown to ACME if this was an outside attacker or an ill-willed employee from WBS. Because of this, ACME has disconnected the B2B VPN with WBS and temporarily discontinued service with WBS until the issue is resolved. They have agreed to fulfill all outstanding orders in the interim. Since WBS has very little technical expertise, they called in ABC Security Consulting Services (ABC) for a thorough analysis of the alleged compromise.
Using risk analysis concepts, we characterize the attack into three compromise scenarios (Table 5) for the purpose

of asset (AV) and Underlying Exposed Assets (UEA) valuation.

Table 5. Three compromise scenarios

One	WBS NT 4.0 Web server (AV); WBS NT Domain (UEA)
Two	WBS UNIX-based Web server (AV); old internal WBS database containing inventory data and pricing for customers and suppliers (UEA)
Three	WBS router; Primary supplier ACME's network

6.4 Incidence Analysis

After extensive network and operational analysis by ABC, several key operational deficiencies were uncovered that revealed the inadequacies of the WBS network. Furthermore, ABC found several key security vulnerabilities and design flaws. These vulnerabilities included:

- The original static Web site was on a NT 4.0 server that was several service packs behind. This server ran multiple vulnerable programs. This server was also configured as a part of the NT domain so it had network access to many of the internal devices as if it was on the internal network. There was also no network access control between this server and the internal network.

- The newer dynamic Web site was on a modern Unix server that was relatively current. This was also the only Unix host on the network. This host was connected to an older internal database that contains inventory, availability, and pricing information. This was the contractor Web site and is a large part of the WBS business plan going forward. Even though this host was relatively current, it still had vulnerabilities within Apache and SSH and had FTP and RPC daemons running.

- WBS has a router that had the WEB management interface open on the internal interface (which was still accessible from the internet). This gives away complete router configuration, including VPN information and internal network architecture.

ABC recommended that there is a definite need for WBS to implement IDS technology to monitor the content of each connection. The recommendation is that a host-based IDS be run on the Web servers and a network-based IDS run at the border. See Table 4 for general statistics on how often these types of attacks occur (note: numbers in Table 4 are based on networks that NetSolve manages).

The choice of the IDS implementation scheme will depend on the scheme that provides the best annual loss expectancy and return on investment (ROI). In Section 6.5 we calculate the ALE and ROI to determine this.

6.5 Risk assessment calculations

Procedurally, once the Asset Valuations (AV & UEA) has been conducted and the Exposure Factors (EF & EFS) estimated, the single loss expectancy (SLE) and the Annual Rate of Occurrence (ARO) are then calculated. In general, there are two types of ARO -Site-specific ARO, which is generated from a site of interest, and National ARO, which is computed, based on the analysis of the annual frequency of threats.

The ARO is estimated based on available industry statistics [14]. The asset values were estimated based on current market prices [14]. The calculations were made using the risk formulas listed in Table 2. The results of the calculations are listed on Table 6 in Appendix 8.

6.6. Interpretation of results

The results shows that auto-response affects primary mitigation windows, which has a direct impact on partially reducing the Annual Rate of Occurrence (ARO). This is illustrated in table 6, where we see a beneficial conservative reduction in ARO of 50% (highlighted in yellow in the "IDS w/Auto-Response" rows for each of the three scenarios). Equally, incident response affects the secondary mitigation window, which impacts exposure factor (EF) and secondary exposure factor (EFS), which in turn impacts the Cascading Threat Multiplier (CTM). This is also illustrated in table 6, where we see a beneficial conservative reduction in EF and EFS of 25% respectively (highlighted in yellow in the "IDS w/ Auto-Response & Incident Response" rows for each of the three scenarios).

These reductions have positive effects on the Annual Rate of Occurrence (ARO) and the ultimately the Return on Investment (ROI) of IDS. Once the aggregate annualized savings (ALE1 - ALE2 or ALE1 - ALE3) occurring from IDS deployment equals the support costs associated to the deployment a positive ROI should materialize. In the case of WBS, the two ROIs (ROI1 & ROI2) for each support profile are as follows:

- Single support with IDS using auto-response (ROI1) = -4%;
- Single support with IDS using auto-response and incident response (ROI2) = 36%;
- MSSP support with IDS using auto-response (ROI1) = 81%; and

- MSSP support with IDS using auto-response and incident response (ROI2) = 155%.

The single support management scheme refers to the management by a single skilled in-house technician, management in which there are five shifts of skilled technicians providing 24x7x365 coverage, while the MSSP schemes refers to the management provided by an MSSP.

7 Conclusion

This studies presented in this paper underscores the importance of the new concepts we have introduced into risk analysis formulas. When an IDS device is deployed in a complex environment a lot of factors are brought to bear on the performance index. In order to accurately measure the performance of the IDS using the annual loss expectancy (ARO) as a measure, it is necessary to formulate the analytical framework for asset valuation and risk calculations. This can be realized using the new concepts and formulas we have proposed.

Because the main function of the IDS in enterprise systems is to restrain or at least mitigate losses resulting from attacks, there is the need to optimize the effectiveness of the IDS using proven deployment techniques. In this regard, this study demonstrates the effectiveness of the proactive deployment technique in mitigation threat occurrence.

Finally, for an effective assessment of the IDS in complex and interdependent environments, there is the need to develop a suitable risk analysis framework. In the end, to maximize the performance of the IDS, you need to have a sound understanding of the enterprise environment including, at a minimum, how it does business, how its connected, where the asset value really lies and what vulnerabilities and associated threats (equating to risk and exposure) need to be analyzed and addressed through sound a security policy and risk mitigation techniques.

References

[1] http://itmanagement.earthweb.com/columns/article.php/1025311

[2] Iheagwara C. and Blyth A, " Evaluation of the performance of IDS systems in a switched and distributed environment," *Computer Networks*, 39 (2002) 93-112

[3] Iheagwara c, Blyth A., Singhal M., "A Comparative Experimental Evaluation Study of Intrusion Detection System Performance in a Gigabit Environment," *Journal of Computer Security*, Vol 11(1), January, 2003

[4] Richards K, "Network Based Intrusio n Detection: a review of technologies," *Computers & Security*, 18 (1999) 671-682.

[5] http://www.usfca.edu/fac-staff/morriss/651/tech_projects/VLAN/benefits.htm

[6] Gilbert, I.E. "Guide for Selecting Risk Analysis Tools." NIST Special Publication 500-174. October 1989. URL: http://csrc.nist.gov/publications/nistpubs/500-174/sp174.txt (22 March 2002).

[7] Venter, H.S., Labuschagne, L., Eloff, J.H.P. "Real-time Risk Analysis on the Internet." March 1999. URL: http://csweb.rau.ac.za/ifip/workgroup/docs1999/11_sec1999.doc (22 March 2002).

[8] Changduk, J., Han, I., Bomil, S. "Risk Analysis for Electronic Commerce Using Case-Based Reasoning." 1999. URL: http://ajis.kaist.ac.kr/download/inter_jnl012.pdf (22 March 2002).

[9] Craft, R., Wyss, G., Vandewart, R., Funkhouser, D. "An Open Framework for Risk Management." 21st National Information Systems Security Conference Proceedings. October 1998. URL: http://csrc.nist.gov/nissc/1998/proceedings/paperE6.pdf (22 March 2002).

[10] Labuschagne, L., Eloff, J.H.P, "Risk Analysis Generations – The Evolution of Risk Analysis." August 1999. URL: http://csweb.rau.ac.za/deth/research/articles/ra_generations.pdf (22 March 2002).

[11] Harris, S., " CISSP Certification Guide" McGraw-Hill/Osborne, 2001, pp72-91.

[12] Brewer, Dr. David. "Risk, Security and Trust in the Open World of E-Commerce." May 1999. URL: http://www.itsecurity.com/papers/p35.htm (22 March 2002).

[13] Brewer, Dr. David. "Risk Assessment Models and Evolving Approaches." IAAC workshop, London. July 2000. URL: http://www.gammassl.co.uk/topics/IAAC.htm (22 March 2002).

[14] Kevin, T., Kinn, D., " Justifying the expense of IDS." LinuxSecurity article, (28 August 2002).

[15] http://www.silicondefense.com/software/acbm/speed_of_snort_03_16_2001.pdf

The net positive effect on the ROIs are based on the aggregate annualized savings from deploying and effectively managing the IDS technology and the resulting impact the IDS technolo gy could reasonably have on the combined effect of the three compromise scenarios described in Table 5.

Appendixes

Appendix 1.

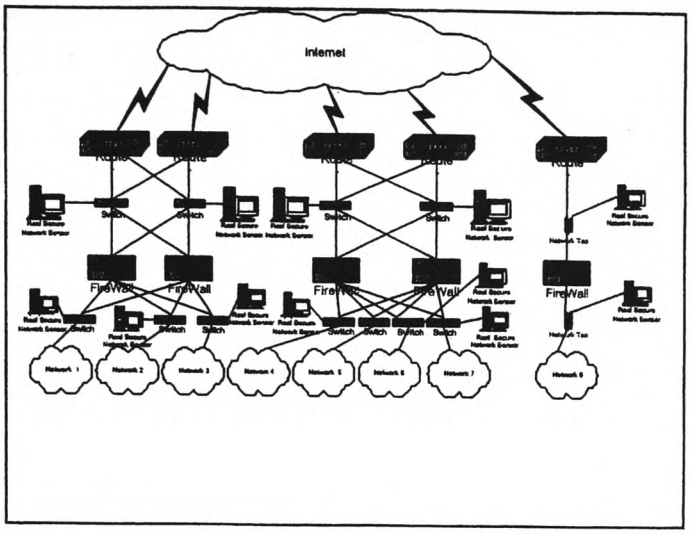


Fig. 1 **Multiple IDS deployment in a large enterprise system**

Appendix 2

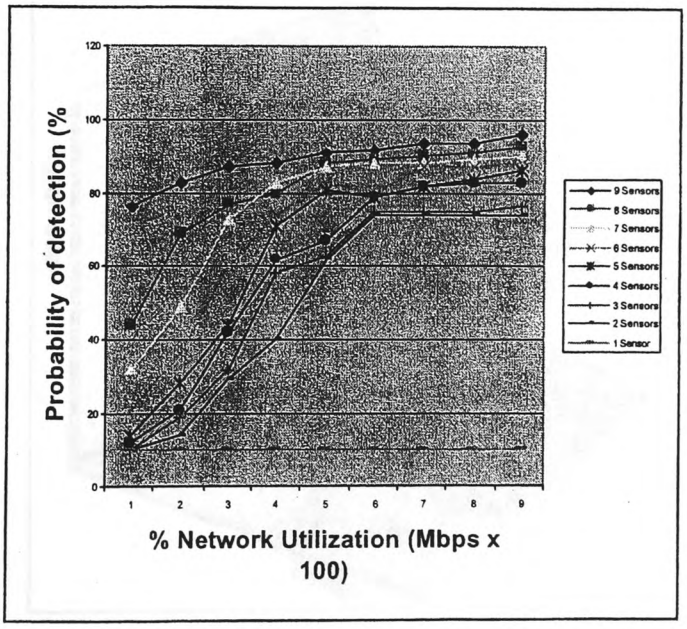


Fig. 2 Probability of detection vs. percent utilization with RealSecure Multiple sensors [3]

Appendix 3

Table1. Example of a Qualitative Analysis questionnaire [11]

Respondent	Severity of Threat	Probability of Threat Taking Place	Potential Loss to the Company	Effectiveness of Firewall	Effectiveness of Intrusion Detection System	Effectiveness of Honey pot
IT Manager	4	2	4	4	3	2
Database Administrator	4	4	4	3	4	1
Application Programmer	2	3	3	4	2	1
System Operator	3	4	3	4	2	1
Operational Manager	5	4	4	4	4	2
Results	3.6	3.4	3.6	3.8	3	1.4

Appendix 4

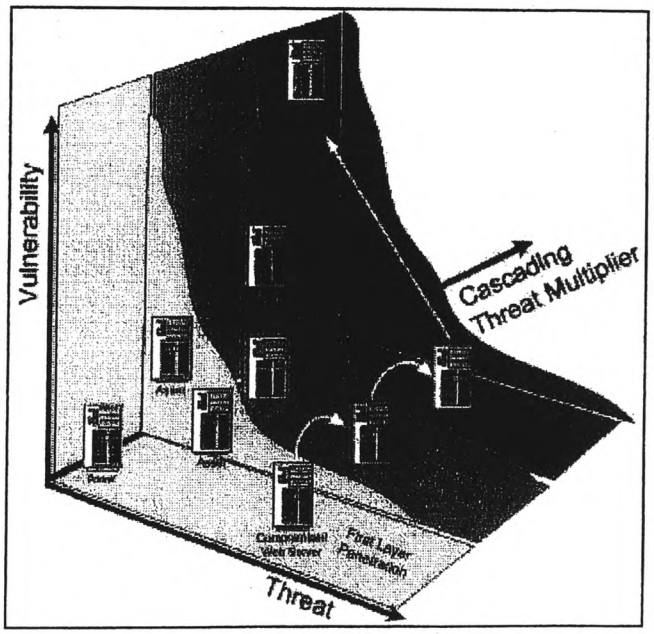


Fig. 3 Cascading threat multiplier

Appendix 5

Table 2. Risk analysis equations

Variable	Formula or expression
Asset Value (AV)	AV = hardware + comm. software + proprietary software + data
Exposure Factor (EF)	EF is the % estimation of the exposure of the initial compromised asset
Underlying Exposed Assets (UEA)	UEA is the estimation of the $ value of the assets behind the compromised in itial asset
Secondary Exposure Factor (EFs)	EFs is the % estimation of the exposure of the UEAs
Cascading Threat Multiplier (CTM)	CTM = 1 + ((UEA x EFs) / AV)
Single Loss Expectancy (SLE)	SLE = EF x AV x CTM
Annual Rate of Occurrence (ARO)	ARO is estimated number, based on available industry statistics or experience
Annual Loss Expectancy **without** IDS (ALE1)	ALE1 = SLE x ARO
Annual Loss Expectancy with IDS using auto-response (ALE2)	ALE2 = conservative 50% reduction of ARO when IDS is managed skillfully with auto-response
Annual Loss Expectancy with IDS using auto-response & incident response (ALE3)	ALE3 = conservative 25% reduction of EF & EFS when IDS is managed skillfully with auto-response and incident response
Annual Cost (T) of IDS Technology and Mgmt	T
Annual Recovery Cost ('R) from Intrusions without IDS	R = ALE1
Annual Dollar Savings (E) gained by stopping intrusions with IDS	E = ALE1 - (ALE2 or ALE3)
Traditional Return on Security Investment (ROSI) equation	ROSI = R - ALE, where ALE = (R-E) + T
ROI of IDS with auto-response (ROI1)	ROI1 = ALE1 - ((ALE1 - (ALE1 - ALE2)) + T)
ROI of IDS with auto-response & incident response (ROI2)	ROI2 = ALE1 - ((ALE1 - (ALE1 - ALE3)) + T)

ppendix 6

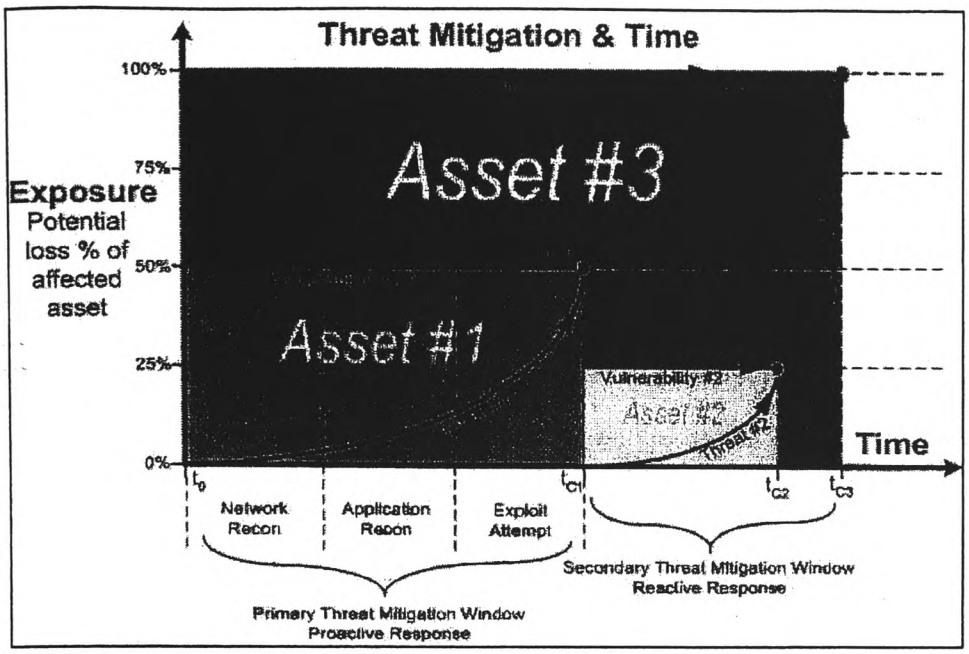


Fig. 5 Threat mitigation & time

ppendix 7

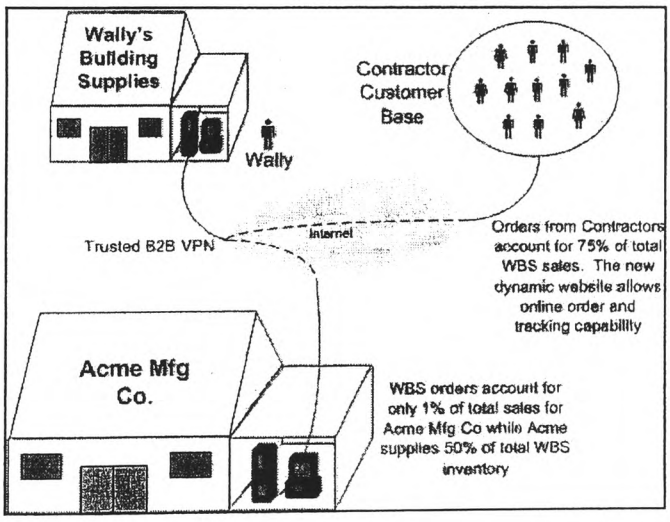


Fig. 6 Wally's building supplies

Appendix 8

Table 6. Risk calculation results the different deployment schemes

Scenario	ROI Scope	AV	EF	ULA	EFS	CTM	SLE	ARO	ALE1 (no IDS)	ALE2 (w/ AR)	ALE3 (w/ AR & IR)	Single Support ROI $	Single Support ROI %	MSSP Support ROI $	MSSP Support ROI %
One	no IDS	$2,000	75%	$20,000	75%	8.5	$12,750	3	$38,250	N/A	N/A	N/A	N/A	N/A	N/A
	IDS w/ Auto-Response	$2,000	75%	$20,000	75%	8.5	$12,750	1.5	$38,250	$19,125	N/A	-$64,092	-77%	-$25,092	-57%
	IDS w/ Auto-Response & Incident Response	$2,000	56%	$20,000	56%	6.6	$7,453	1.5	$38,250	N/A	$11,180	-$56,147	-67%	-$17,147	-39%
Two	no IDS	$20,000	50%	$50,000	50%	2.3	$22,500	2	$45,000	N/A	N/A	N/A	N/A	N/A	N/A
	IDS w/ Auto-Response	$20,000	50%	$50,000	50%	2.3	$22,500	1	$45,000	$22,500	N/A	-$60,717	-73%	-$21,717	-49%
	IDS w/ Auto-Response & Incident Response	$20,000	38%	$50,000	38%	1.9	$14,531	1	$45,000	N/A	$14,531	-$52,748	-63%	-$13,748	-31%
Three	no IDS	$3,000	75%	$200,000	50%	34.3	$77,250	1	$77,250	N/A	N/A	N/A	N/A	N/A	N/A
	IDS w/ Auto-Response	$3,000	75%	$200,000	50%	34.3	$77,250	0.5	$77,250	$38,625	N/A	$41,592	54%	$5,592	13%
	IDS w/ Auto-Response & Incident Response	$3,000	56%	$200,000	38%	26.0	$43,875	0.5	$77,250	N/A	$21,938	-$27,905	-34%	$11,096	25%
WBS ROI with IDS Auto-Response (ROIi)									$160,500	$80,250	N/A				
WBS ROI with IDS Auto-Response & Realtime Incident Response (ROIr)									$160,500	N/A	$47,648				

Appendix 8

"The Effect of Intrusion Detection Management Methods on the Return on Investment"
Computers & Security Journal, vol 23, 213-228

Computers & Security (2004) 23, 213–228

ELSEVIER

Computers & Security

www.elsevier.com/locate/cose

The effect of intrusion detection management methods on the return on investment

Charles Iheagwara*

Information Technology Security Department, Una Telecom, Inc., 4640 Forbes Boulevard, Suite 200, Lanham, MD 20706, United States

Received 9 May 2003; revised 15 September 2003; accepted 24 September 2003

KEYWORDS
Intrusion detection systems;
Return on investment;
Cost-effectiveness;
Cost–benefit analysis

Abstract This paper examines how implementation methods, management methods, and Intrusion Detection System (IDS) policy affect Return on Investment (ROI). The paper will seek to demonstrate the value associated with a well thought out implementation and effective lifecycle management of IDS technology and will culminate in a case study with a number crunching exercise to calculate the ROI for an IDS deployment by a hypothetical financial company named UTVE, Inc. on risk.

The paper also discusses general IDS types and expands on the impact that the logical location of a company's critical networked assets could have on the risk equations. To this end, the Cascading Threat Multiplier (CTM) is introduced to expand on the Single Loss Expectancy (SLE) equation. Also, implementation and management costs based on various support profiles and commonly accepted risk equations are reviewed. Finally, a formula for calculating ROI for security, otherwise commonly known as Return on Security Investment (ROSI) is devised.

Introduction

An IDS is a security system that monitors computer systems and network traffic and analyzes that traffic for possible hostile attacks originating from outside the organization and also for system misuse or attacks originating from inside the organization. By providing information to site administration, an IDS allows not only for the detection of attacks explicitly addressed by other security components (such as firewalls and service wrappers), but also attempts to provide notification of new attacks unforeseen by other components. Intrusion detection systems also provide forensic information that potentially allows organizations to discover the origins of an attack.

Currently there are two basic approaches to intrusion detection. The first approach is to define and characterize correct static form and/or acceptable dynamic behavior of the system and

* Tel.: +1-301-459-7674; fax: +1-202-659-2810.
E-mail address: iheagwarac@aol.com.

0167-4048/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cose.2003.09.006

then to detect abnormal behavior by defining statistical relations. This is called *anomaly detection*. It relies on being able to define the desired form or behavior of the system and then to distinguish between that definition and undesirable form or anomalous behavior. While the boundary between acceptable and anomalous forms of stored code and data can frequently be precisely defined, the boundary between acceptable and anomalous behavior is much more difficult to define.

The second approach, called *misuse detection*, involves characterizing known ways to penetrate a system, usually described as a pattern, and then monitoring for the pattern by defining rule-based relations. The pattern may be a static bit string, for example, a specific virus bit string insertion. Alternatively, the pattern may describe a suspect set or sequence of events.

Intrusion detection systems have been built to explore both approaches: anomaly detection and misuse detection. In some cases, they are combined in a complementary way in a single intrusion detector. There is a consensus in the community that both approaches continue to have value.

As a matter of practical reality, organizations evaluate the effectiveness of the IDS implementations from both technical and economic perspectives. Thus, the overall evaluation of any IDS implementation is based on a wide range of criteria especially when a choice has to be made as to what is the right IDS product.

Practically speaking, a very important but often neglected facet of intrusion detection is its *cost-effectiveness*, or *cost–benefit* trade-off. An educated decision to deploy a security mechanism such as IDS is often motivated by the needs of security risk management. The objective of IDS is therefore to provide protection to the information assets that are at risk and have value to an organization. An IDS needs to be cost-effective because it should cost no more than the expected level of loss from intrusions. This requires that an IDS should consider the trade-off among cost factors, which at the minimum should include development cost, the cost of damage caused by an intrusion, the cost of manual or automatic response to an intrusion, and the operational cost, which measures constraints on time and computing resources.

It should also be noted that it is not always necessary to justify the cost of an organization's IDS deployment because the implementation might be undertaken as part of a standard due care.

The performance of IDS for many organizations is not just measured in the ability of the IDS to capture or prevent attacks but on its value when expressed in economic terms. This is more so because when choosing a security product, companies tend to justify their investments based on both economic returns and technical performance. In the selection of an IDS product, performance is measured using such factors as scalability, availability, ROI and the total cost of the system relative to the price of the system the IDS is protecting, just to mention a few.

A positive ROI of an IDS is dependent upon an organization's deployment strategy and how well the successful implementation and management of the technology helps the organization achieve the tactical and strategic objectives it has established. For organizations interested in quantifying the IDS's value prior to deploying it, their investment decision will hinge on their ability to demonstrate a positive ROI. The ROI has traditionally been difficult to quantify for network security devices, in part because it is difficult to calculate risk accurately due to the subjectivity involved with its quantification. Also, business-relevant statistics regarding security incidents are not always available for consideration in analyzing risk.

In considering an implementation of an IDS technology, a positive ROI can be understood by analyzing the difference between Annual Loss Expectancy (ALE) without IDS deployment and the ALE with IDS deployment, adjusted for technology and management costs. The ultimate initial goal, then, should be to prove that the value proposition (re: a benefit in the form of a quantifiable reduction in ALE) in implementing and effectively managing the IDS technology is greater than the implementation and management costs associated with deploying the IDS technology.

The insights gained from previous research studies that describe proven techniques to implement the IDS technology could be helpful. However, it is important to note that there are no known research studies on the ROI of IDS. Related works border on IDS performance and cost models, none of which integrated or established a link between the technical, operational and cost/economic factors that serves as a gauge for justifying IDS deployment.

The related works are fundamental studies on IDS performance (Iheagwara and Blyth, 2002; Iheagwara et al., 2003; Richards, 1999) that treat the relationship between deployment techniques and attack system variables and the performance of the IDS; and models on cost–benefit/sensitive analysis (Irvine et al., 1999; Lee et al., 1999) for intrusion detection deployment.

Richards (1999) evaluates the functional and performance capabilities of the industries' leading commercial type IDS. In the areas tested, the

performance of the IDS was rated based on their distinctive features, which were characterized into different performance indexes. The research work represented a new direction for IDSs in that it moved the focus away from scientific concepts research to performance evaluation of the industries' best products. However, the study was limited to a small proto design isolated to a non-switched network, which did not reveal the impact of packet switching on the accuracy and ability to capture attack packets in their entirety. Iheagwara and Blyth (2002) expand this effort to an evaluation study of the effect of deployment techniques on IDS performance in switched and distributed system. They demonstrated that monitoring techniques could play an important role in determining the effectiveness of the IDS in a switched and distributed network.

Iheagwara et al. (2003) in a comparative experimental evaluation study of intrusion detection system performance in a gigabit environment examine the system benefits of using a single Gigabit IDS sensor instead of multiple Megabit sensors for a wide range of defined system attacks, network traffic characteristics, and for their contexts of operational concepts and deployment techniques.

As mentioned above, cost–benefit model and analysis studies of IDS deployments are relatively few. Lee et al. (1999) study the problem of building cost-sensitive intrusion detection models. For intrusion detection, Irvine et al. (1999) define auditing of network control functions in intermediate nodes, and rule-based network intrusion systems in the total subnet as the mechanisms. They also discuss the costs of those security services and mechanisms.

In contrast to the above, the focus of this paper is the return of investment of IDS products and an examination of how implementation methods, management methods, and IDS policy affect the ROI. The paper will seek to demonstrate the value associated with a well thought out implementation and effective lifecycle management of IDS technology and will culminate with a number crunching exercise to calculate the ROI for an IDS deployment by a hypothetical brick and mortar wholesale hardware supply company named UTVE, Inc. on risk.

The rest of the paper is organized as follows. The current IDS implementation is discussed in the following section. Then in the next sections, the management and costs structures, the Cascading Multiplier Effect (CTM) and a discussion on the effects of proactive and reactive management techniques are given. This is followed by a case study on the ROI and finally a conclusion section is presented.

Current IDS implementation

In enterprise systems, IDS implementation requires deployment of the IDS either at the computer system that is the putative target or placement on a network level where traffic can be evaluated or where information aggregated from various hosts can give insight in coordinated attack scenarios.

Hence it is important to maximize the implementation through effective deployment techniques. Ptacek and Newsham (1999) and Iheagwara and Blyth (2002) conduct studies to evaluate the effect of deployment techniques on the performance of the IDS. The studies demonstrate that the IDS can be very effective if optimally deployed or it could just be another waste for the company if improperly managed. Since the IDS effectiveness in detecting intrusions depends as much on the deployment technique, a significant change in the approach to the implementation of intrusion detection is needed for improvements.

Of interests are IDS product implementation technology and the architecture that has so often been used to evaluate their effectiveness. The next two sections discuss the technologies and evaluation architectures.

Technologies

Intrusion detection is an overlay of two separate and different technologies: Network IDS (NIDS) and Host-based IDS (HIDS) systems. The primary advantage of NIDS is that it can watch the whole network or any subsets of the network from one location. Therefore, NIDS can detect probes, scans, and malicious and anomalous activity across the whole network. These systems can also serve to identify general traffic patterns for a network as well as aid in troubleshooting network problems. When enlisting auto-response mechanisms, NIDS can protect independent hosts or the whole network from intruders. NIDS does, however, have several inherent weaknesses. These weaknesses are its susceptibility to generate false alarms, as well as its inability to detect certain attacks called false negatives. NIDS also is not able to understand host specific processes or protect from unauthorized physical access. HIDS technology overcomes many of these problems. However, HIDS technology does not have the benefits of watching the whole network to identify patterns like NIDS does. A recommended combination of host and network intrusion detection systems, in which an NIDS is placed at the network border and an HIDS is deployed on critical servers such as databases,

Web services and essential file servers, is the best way to significantly reduce risk.

Implementation architecture

Generally speaking, most (host- and network-based) intrusion detection systems have common architectures, meaning that most host/network systems work as agents reporting to a central console. The present implementation architecture is built on the concepts of the qualitative and quantitative operational functionality of the IDS. In Fig. 1 the IDS architecture is presented into the quantitative and qualitative evaluation modules.

In the quantitative evaluation module the main components—detection engine, filters, alerting and configuration facilities—are used for functional and performance testing of the IDS. The IDS detection engine's (sensor's) job is to watch the network and detect attacks, a role that is performed by the packet-processing engine. To do this, the sensor looks at every packet on the network it is watching. The busier the network, the more packets there are to watch. If the sensor cannot keep up, it will start to miss (or drop) packets. In the case an attack spans multiple packets, the sensor holds the packets, assembles them and makes a determination on whether there is an attack. The extent and scope of accomplishing the above roles is a gauge of the effectiveness of the IDS and that is why the IDS performance is evaluated based on the ability of the processing engine to effectively filter and reassemble packets to any given network throughput.

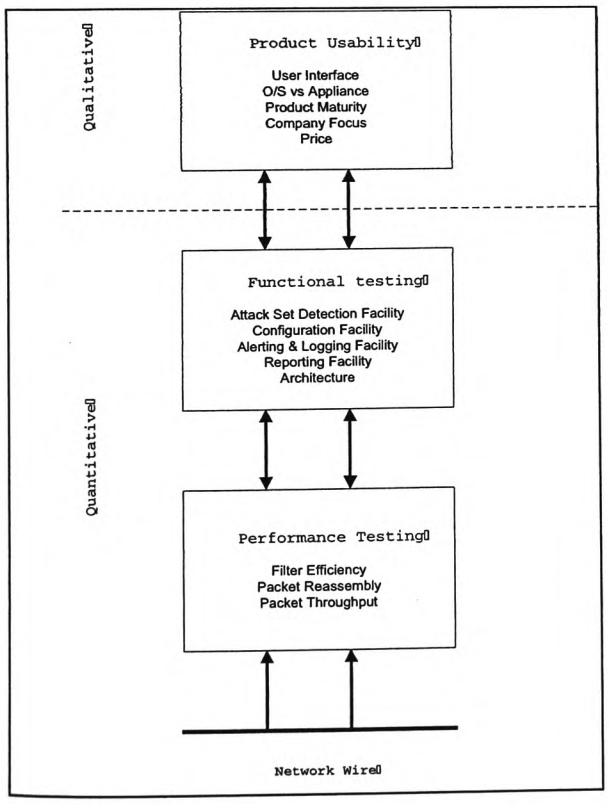


Figure 1 The standard IDS architecture.

The quantitative architecture includes attack set detection, configuration alert triggering, logging and reporting facilities modules that define the ability of the IDS to accurately characterize the operational setup of the environment and customizable utilities.

The qualitative architecture includes a module that defines the product usability effectiveness of the IDS based on certain usability features such as the ease of user interface (ease of use, ease of configuration, ease of filter customization); integration and interoperability with operating systems and existing network infrastructure; product maturity; company focus and price.

The associated management and costs structures of IDS implementation are presented in the following section.

Management and costs structures

Deployment cost

The associated cost of HIDS deployments can vary depending on vendor and software versions. A good baseline is that agents can cost between $500 and $2000 each and consoles may cost in the $3000–$5000 (Kevin and Kinn, 2002) range. This does not include OS, hardware or maintenance costs. Network intrusion detection systems can be deployed as stand-alone hosts with a possible management interface or distributed sensors and management console. Generally speaking, commercially available sensors run in the $5000–$20,000 range (Kevin and Kinn, 2002) depending on vendor, bandwidth and functional capabilities. Management consoles can be free or can cost several thousand dollars depending on the vendor. This does not necessarily include hardware or back-end databases. IDS usage requires human interaction at the end point because the IDS will generate pertinent information and data, but this serves no purpose without subsequent examination of the data. This will certainly require allocating a skilled staff for IDS management, log analysis, etc. If this is not the case, then the investment will fail to pay off.

The total cost of an IDS deployment depends on implementation costs combined with the costs for

Table 1 Cost of individual components

Expense	Value
Network IDS	$10,000
Host IDS	$1000
NIDS/HIDS management station	$5000 (for some products)
Maintenance	15% of the cost of IDS
MSSP network IDS management per year	$24,000 ($2K per month)
MSSP host IDS management per year	$6000 ($500 per agent per month)
Engineer cost	$75,000 ($60,000 salary plus $15K benefits & administration)
Group manager cost	$100,000 ($80,000 salary plus $20K benefits & administration)

managing the technology. Some standard implementation and management methods common to IDS deployments include using a Managed Security Services Provider (MSSP), utilizing a single in-house employee or technician, or enabling $24 \times 7 \times 365$ multi-shift coverage in-house with a skilled technical staff. Of course the size of the organization and its associated IT budget (or lack thereof) factor in to how the IDS technology will be deployed and managed. The generalized cost structure used for discussions in this paper is shown in Table 1.

Comparison of aggregate costs of different implementation schemes

Based on this generalized cost structure in Table 1, let us now consider the aggregate costs of three different IDS deployments. Tables 2 and 3 represent implementation (purchase) costs combined with life cycle management costs over a three-year period. The three scenarios include management by a single skilled in-house technician, management in which there is five shifts of skilled technicians providing $24 \times 7 \times 365$ coverage, and management provided by an MSSP. It is very important to understand that full-service MSSPs will provide $24 \times 7 \times 365$ coverage just like the multi-shift internal coverage provides. For completeness, there will be a review of two different IDS deployments (one small and one medium) and the cost structure of implementing and managing them.

From the numbers it is evident that in smaller IDS deployments the value proposition of MSSP support is very strong relative to internal $24 \times 7 \times 365$ multi-shift support. In larger IDS deployments, the cost differential between internal (highly skilled) multi-shift coverage and MSSP coverage diminishes due to economies of scale on the internal multi-shift coverage side. Single support coverage is not a realistic option to consider when contemplating a deployment of 30 security devices. Also, this cost model does not take into account proprietary tools development necessary to manage several different types of technology (if that were the case) effectively.

The concept of cascading threat multiplier

The main concepts of risk management and related equations are given in Appendix 3. In the world of new technologies, the interplay of technological processes, policies and risk management introduces complexities into risk management and requires the development and introduction of a new concept—Cascading Threat Multiplier (CTM) used to accurately calculate the ROI for any acquired or developed technology.

Consequently, the introduction of a new concept, Cascading Threat Multiplier (CTM), into the mix will help in any analytical discussion that helps in distilling a meaningful ROI calculation in order to determine the effectiveness of deploying IDS technology into a given network.

The Cascading Threat Multiplier (CTM) is a multiplying factor that will be included into our

Table 2 Implementation and management cost of one network IDS and two host IDS

	Single support	$24 \times 7 \times 365$ Multi-shift support	MSSP support
Technology cost ($)	24,650	24,650	24,650
Management cost ($)	225,000	1,425,000	108,000
Total cost ($)	249,650	1,449,650	132,650
Average cost per year ($)	83,217	483,217	44,217
Average cost per device per year ($)	27,739	161,072	14,739

Table 3 Implementation and management cost of 15 network IDS and 15 host IDS

	Single support	24 × 7 × 365 Multi-shift support	MSSP support
Technology cost ($)	N/A	268,250	268,250
Management cost ($)	N/A	1,425,000	1,350,000
Total cost ($)	N/A	1,693,000	1,618,250
Average cost per year ($)	N/A	564,417	539,417
Average cost per device per year ($)	N/A	18,814	17,981

expanded definition of Single Loss Expectancy (SLE). CTM is somewhat subjective and is introduced mainly for the purpose of adding a little more "flavor" to SLE. CTM factors in the importance of other critical assets tied (re: networked) to the specific asset being analyzed in the SLE calculation. It also coaxes risk analysts to think in broader terms and to look at the bigger picture when considering the risks associated with the compromise of a given asset. The formula for CTM is as follows:

$$\text{Cascading Threat Multiplier (CTM)} = 1 + ((\text{UEA} \times \text{EFs})/\text{AV})$$

In this formula, Underlying Exposed Assets (UEA) is measured in dollars. These are the assets that are now exposed due to the compromise of a specific asset. Asset Value (AV) is identical to the calculation described elsewhere in this paper. Exposure Factor (EFs) represents secondary exposure factor and is related to the percentage loss on the UEAs. Secondary Exposure Factor (EFs) is very similar to Exposure Factor (EF), as described in the standard equation in Appendix 3, with a few minute differences.

It should be noted that the CTM formula is relevant only to the threat of unauthorized disclosure, as would typically be of primary concern to a military or diplomatic agency. It does not address interruption in service, as would for instance be very important for an Internet merchant. Likewise, it does not address data corruption, as would for instance be very important for a bank.

The primary reason for introducing EFs is to factor in the importance of an asset's logical location within a network. For example, if the asset is a Web server that is in a true Demilitarized Zone (DMZ) and has no access to the network or to any other corporate servers, EFs would be low since it is unlikely that an attacker can use this device to further compromise the network. But if the asset is on the same broadcast domain as other servers are (such as e-mail, DNS and FTP), or there is no access control between the asset and other servers, then EFs will be higher. Finally, if the asset is on a network that has access to the rest of the network, the Secondary Exposure Factor (EFs) will be very high. Examples of this would include hosts that offer some public services but are terminated within the internal network or hosts that have valid Secure Shell (SSH) keys to all hosts. SSH is a protocol used to provide strong authentication and secure communications over unsecured channels.

It is important to consider what assets are easily (or even not so easily) accessible from a specific networked asset once that asset is compromised. When a given asset is compromised and used as a staging point for attacks on other assets inside and outside a company's network, it could have potentially devastating consequences for the organization. If an attack is staged from the compromised asset to another asset outside the organization, even when the owner was not directly involved in the malicious activity, they can and probably will be held accountable. One can envision the UEA factor of SLE representing some portion of a trusted business partner's assets. It is easy to imagine the negative business impact the offending organization would encounter if one of its compromised assets were used as a staging ground to compromise and damage its business partner's assets.

What is the risk, quantified in dollars, of not considering a business partner's assets when performing a valuation exercise on your company's assets, ones that, if compromised, may enable access to more sensitive data and systems? The CTM concept provides the analytical framework to closely scrutinize the assets under an organization's control, assign more comprehensive valuations to those assets, and to more accurately measure the impact that compromising of these assets could have on the organization.

As a practical example, it is assumed that a Web server has been compromised and used by a malicious person to stage attacks on other networked assets containing critical data valued at 10 times the amount (in dollars) of the data contained on the compromised Web server. As the perpetrator hopscotches his way from asset to asset, penetrating deeper and deeper into the network, he may

finally gain access to critical data on a vulnerable asset deep inside the company's network. The CTM for the Web server would be calculated as follows with a summation (re: best guess) that the Secondary Exposure Factor (EFs) of the Underlying Exposed Asset(s) (UEA) is 70%: CTM = 1+((10 × 0.7)/1) = 8. The CTM has increased the SLE for the compromised Web server by a factor of 8. The white arrow originating from the compromised Web server should be followed to better visualize this concept as shown in Fig. 2.

Tying the CTM concept back into the SLE calculation, a new definition of Single Loss Expectancy can be expressed as:

$$SLE = EF \times AV \times CTM$$

A thorough risk management exercise should factor in the CTM concept by executing a more comprehensive valuation methodology that included more subjective, intangible factors into their Asset Value (AV) variable calculation. As mentioned above, goodwill (i.e. business and consumer loyalty built on trust) and opportunity costs (i.e. choosing not to consider the effect that a compromised asset can have on other assets) are somewhat analogous to the CTM concept when these intangibles are factored into the Asset Valuation (AV) used in the SLE calculation.

The importance of capturing intangible value, and understanding the risks associated with jeopardizing the value, is one of the more challenging aspects of risk and return analysis. Introducing the CTM concept into the traditional SLE calculation will make the capture of the intangible aspects of asset valuation a little less daunting of a task.

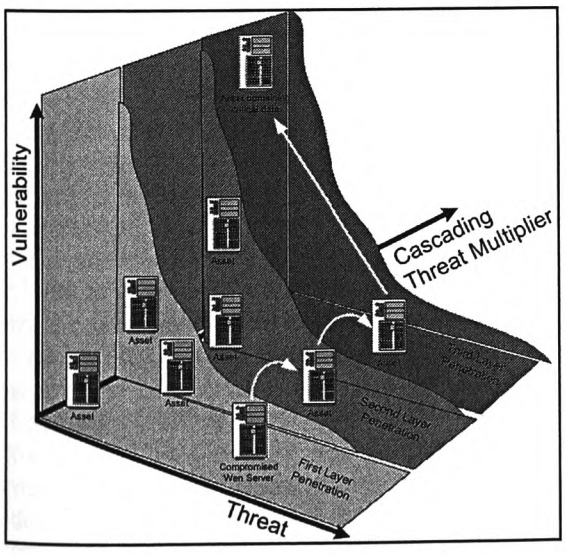


Figure 2 Cascading threat multiplier.

Finally, the risk analysis calculations listed above can be tied to an accepted formula for calculating the ROI for a security product.

$$ROI = \text{recovery cost } (R) - ALE$$

where $ALE = (R - E) + T$, and E equals the savings gained by preventing an attack and T equals the cost of a security product.

It should be noted that the use of the ALE approach in risk management is not suitable when a worst-case analysis involving discussion of reputation risk and loss of goodwill is required.

Section "The effects of proactive vs. reactive management on risk" is a discussion on proactive and reactive management methodology and a demonstration on how this methodology affects analysis of risk. This will set up the framework for the calculation of IDS ROI and will culminate in a case study in section "A case study on IDS ROI calculation modular approach" to demonstrate the efficacy of the new concept (CTM). Finally, all the numbers will be tied together to demonstrate the devised technique for calculating the ROI for UTVE deployment of one network-based IDS and two host-based IDSs.

The effects of proactive vs. reactive management on risk

Independent of implementation and management costs, the method in which the devices are managed can have a serious effect on the ROI. As a result, the key question to answer is: is the system going to be proactive or reactive as security events are detected? Table 4 depicts the normal event flow in each method. A proactive implementation response is automated by the system while a reactive implementation response is manually driven with the help of enlisted personnel.

By examining the Annual Loss Expectancy (ALE = ARO × SLE, where SLE = exposure factor × asset value × cascading threat multiplier), the variables that are affected by each of these two management methods can be determined. In a reactive design, where personnel must be engaged to respond to each event, the exposure factors (primary [EF] and secondary [EFs]) will be affected. In a proactive design there will be similar benefits to the exposure factors (re: a reduction) and, in addition, the ARO will be influenced in a beneficial way as well. It will be beneficial to use the concept of primary and secondary mitigation windows to demonstrate the impact of threat vs. time. As illustrated in Appendix 1, the primary mitigation

Table 4 Proactive and reactive management methods

Method	System actions	Personnel actions	Follow up information
Reactive	Log → alert → respond	Respond → analyze → eradicate	Forensics and evidence
Proactive	Respond → log → alert	Analyze → eradicate if necessary	Forensics and evidence

window affects ARO while the secondary mitigation window affects exposure factor and cascading threat multiplier. An effective way of impacting ARO is through automated response.

Auto-response can take many forms. On host-based IDS this is sometimes called shielding, where a specific process is terminated. Network-based IDS generally employs TCP resets or shunning. TCP resets effectively kill one specific session based on suspicious activity, but it still allows other activity from that same IP. Shunning, on the other hand, changes firewall rules or router access lists and effectively denies all traffic from that host for a specific period of time. In essence, shielding will protect a single host from one process, resets will protect a host from a specific session, and shunning will protect the entire network from a specific host for a pre-determined amount of time.

The accuracy of automated response can vary tremendously. This is dependent on the skill level of the engineers managing the devices. If the engineers are moderately skilled then auto-response will not be very effective, which may adversely affect the ROI of the IDS deployment. This adverse effect may manifest itself in the form of a loss of productivity from network-related problems due to improperly implemented auto-response, as well as the additional fallout related to a false sense of security throughout the company.

With skilled engineers managing the devices, auto-response can be very accurate and effective. The data generated over a period of 30 days from the network of NetSolve, Incorporated (Kevin and Kinn, 2002) will be used to illustrate the accuracy of automated response. If Code Red and Nimda activities are included, in 99.96% of the attacks, where automated response was used to mitigate the threat, the activity was malicious. Excluding large-scale worms, the attacks were malicious in 95.8% of auto-response uses. Of the 4.2% of the traffic that was not malicious, not all of it was desirable. Some of this traffic was peer-to-peer programs, on-line gaming, chat and other undesirable traffic that triggered alarms. The percentage of traffic that was denied and business related was very small. It should be noted that many of these devices provide numerous different techniques for ensuring that very little, if any, legitimate traffic is denied through the use of automated response.

The most recent statistics (http://www.silicondefense.com/software/acbm/speed_of_snort_03_16_2001.pdf) was used to determine how effective the device is in recognizing attacks. In this test the worst NIDS detected 67 of 109 attacks or 61.5%, while the best detected 94 of 109 attacks for an 86.2% detection rate. Even the worst case, the 61.5% detection rate, was out of the box (http://www.nss.co.uk/Articles/IntrusionDetection.htm) and it was reported that it would not be difficult to improve this with some custom signatures and tuning.

The above could be interpreted to mean that the worst IDS tested can still detect at least 61.5% of attacks. Realistically that number should be closer to 70% when a skilled engineer or technician manages the device. The auto-response feature, when properly used, can be a very effective method of reducing the ARO. This provides some general numbers, which can be plugged into the equations for calculating a ROI for UTVE.

A case study on IDS ROI calculation modular approach

The lack of established literature on a suitable management approach that can maximize the IDS ROI mandates the use of a case study approach that permits the in-depth exploration of the benefits of performing an ROI analysis to maximize the management techniques of intrusion detection systems. From these, it is possible to glean some general concepts about IDS ROI and determine the viability of the management approach that will enhance the maximization. By developing the examples, it is also hoped to develop a possible method of reasoning about IDS ROI more generally. The case study will be presented in the context of events and risk analysis in a hypothetical company called UTVE, Inc.

Framework for risk analysis and ROI computation

In order to prepare for the IDS ROI computational model, it will be useful to set up a hypothetical company called UTVE and through a case study

present the threat and incidence scenarios to calculate the ROI based on the effective implementation and lifecycle management of HIDS and NIDS technologies. There is the need to articulate a holistic approach and, at the same time introduce some new concepts for analyzing risk. In the analytical discussion leading up to the calculation of the ROI, commonly accepted formulas and definitions associated with asset valuation, exposure, threat, vulnerability and loss expectancy will be used. The Cascading Threat Multiplier (CTM), an additional factor added to the mix, enables the expansion of the risk assessment widely accepted calculation for Single Loss Expectancy (SLE) where, traditionally, SLE = Exposure Factor (EF) × Asset Value (AV).

In order to stress the importance of the intangible considerations that will help to apply a holistic approach for quantifying risk and calculating a meaningful ROI, the concepts of goodwill and opportunity costs should be considered when performing valuation exercises on company assets. Although intangible factors inherently introduce subjectivity into risk and return analysis, it is nonetheless an important step to consider intangibles before one can arrive at a more meaningful calculation of the ROI.

It is worth mentioning here that, in general, it may be safe to assume that organizations would tend to undervalue certain data assets if they have not fully taken into account (or bothered to understand for that matter) how these assets relate to the "big picture". It is simple human nature to take the path of least resistance when given a choice. But that is a very dangerous path to take for anyone attempting to arrive at an accurate assessment of the value of data assets residing on their network.

Understanding the tangible costs and benefits of an asset is much easier than understanding, or even considering for that matter, the intangible costs and benefits associated with that same asset. Clarifying this understanding is a challenge and one that will be addressed throughout the rest of the paper as the IDS ROI is calculated in the case study for UTVE, Inc.

Ultimately, the framework is the use of hypothetical events and data derived from such events to develop a process model for the computation of IDS ROI. The threat events and the incidence analysis are given in the context of risk analysis.

Methodology

The methodology used in the case study takes a pragmatic approach towards the issue of calculating the IDS ROI. First, the enterprise business, IT infrastructure, business relations and security practices are described. This is followed with a discussion on the threats and attacks that compromised the security of the business. In this case, a series of defined and designated attacks to compromise the system are mimicked. Each attack exploits a specific vulnerability in the enterprise network system. Next, the attacks, compromises and contributing factors are analyzed and the sources of the security breach delineated. Part of the analysis is the recommendation of the necessary safeguards to forestall future attacks and in this case deployment of intrusion detection systems. Based on the results of the analysis, the Annual Loss Expectancy (ALE) is quantified, the analytical techniques that are pertinent to the IDS ROI, and their relevance in developing a model for calculating the ROI for IDS deployment in business settings are described. The study will culminate into a discussion of the best management and effective techniques for deploying the IDS to maximize the ROI.

The UTVE enterprises

UTVE's remote offices (Toronto, Manchester, Notting, Sony) are connected via private T1 lines to the corporate office with no Internet outlet. Employees of the firm who require access to company data while out of the office use the VPN over the Internet.

In order to successfully conduct business with UTVE enterprises, customers and business associates need to have consistently reliable telephone, fax, e-mail, Internet, file, print and database access, whether in a remote office or in the corporate office. UTVE sales associates must also have the same system reliability and availability while remotely accessing the corporate systems over the VPN (Virtual Private Network). Remote users are primarily sales associates and trusted business associates who connect to UTVE's VPN over some sort of broadband technology. The remaining associates need VPN access while traveling, which is typically dial-up access.

VPN attack and risk analysis

The UTVE network was compromised when a malicious attacker gained access to UTVE's data and network through the VPN tunnel that was established with one of its business associates. Because of this, UTVE disconnected its VPN connection with

Table 5 UTVE's asset valuation

Asset category	Replacement cost ($)
Accounts payable	816,907.00
Applications	50,000.00
Cash accounts	500.00
Communications hardware	10,000.00
Communications software	4000.00
Databases	15,000.00
Facilities	500,000.00
Hardware	120,000.00
Office equipment	300,000.00
Personnel	5,000,000.00
Procedures	530,000.00
Security	100,000.00
Supplies and consumable	80,000.00
System software	10,000.00

the business associate (ACME) and temporarily halted any business transactions with ACME until the issue is resolved.

After an extensive network and operational analysis of the incidence, several key operational deficiencies were uncovered that revealed several security flaws of the UTVE network. The description of the compromised assets and underlying exposed assets is given in Appendix 2.

The general replacement cost for UTVE's asset category is given in Table 5. It is assumed that the valuation exercise was separately undertaken before the ROI calculations, which in itself, is not the best approach. The replacement cost is the cost of acquiring an asset as itemized in the asset category in the event of loss.

To quantify and characterize the loss associated with the VPN attack, a quantitative risk analysis was conducted. The analysis revealed that data integrity was affected by the intrusion. The various incidents that could be associated with the intrusion at UTVE enterprise are shown in Table 6. The ALE is based on an ARO of three (3.0).

In Table 6, the SLE and ALE are presented for each incident. The ALE is generated by multiplying the SLE for the incident by the ARO of the threat. The overall ALE for a threat is the sum of the ALEs for each of the associated incidents. This is shown as the total of the third column. The percentage of this total represented by the ALE for each incident is indicated in the fourth column. Also shown for each threat is a bar chart that provides a visual presentation of the relative magnitudes of the ALE for each incident. Pie charts are also provided to indicate the percentage of each threat ALE that is accounted for by each incident that is used in its calculation.

The incidence class is the grouping of the various losses in the event of a threat materializing according to their form. For example, the direct loss is the loss from the associated asset and modification is the loss in monetary value due to modification of data, application, system, etc. The concepts and definitions for ALE and SLE have been given in Appendix 3 and the plots are represented in Figs. 3 and 4, respectively.

Risk mitigation

As a result of the risk analysis, the following mitigation measures were recommended.

- Guidelines should be created for the use of UTVE remote access facilities. Example—access privileges are generally only granted to managers, team leaders, associates responsible for overnight/weekend support, and sales staff.
- Employees or those requiring remote access to the network should have the approval of senior management, as well as the operation department.
- There should only be one method of connecting to the network from a remote location. The IT

Table 6 Losses from compromised data integrity with an ARO of three (3)

Incident class	SLE ($)	ALE ($)	% Of total ALE
Direct loss, procedures	18,145	54,435	71.6
Direct loss, security	3596	10,788	14.2
Direct loss, applications	2058	6174	8.1
Modification, databases	513	1540	2.0
Direct loss, databases	427	1282	1.7
Direct loss, system software	416	1248	1.6
Direct loss, communications software	166	499	0.7
Modification, applications	12	37	0.0
Modification, system software	2	7	0.0
Modification, communications software	1	3	0.0

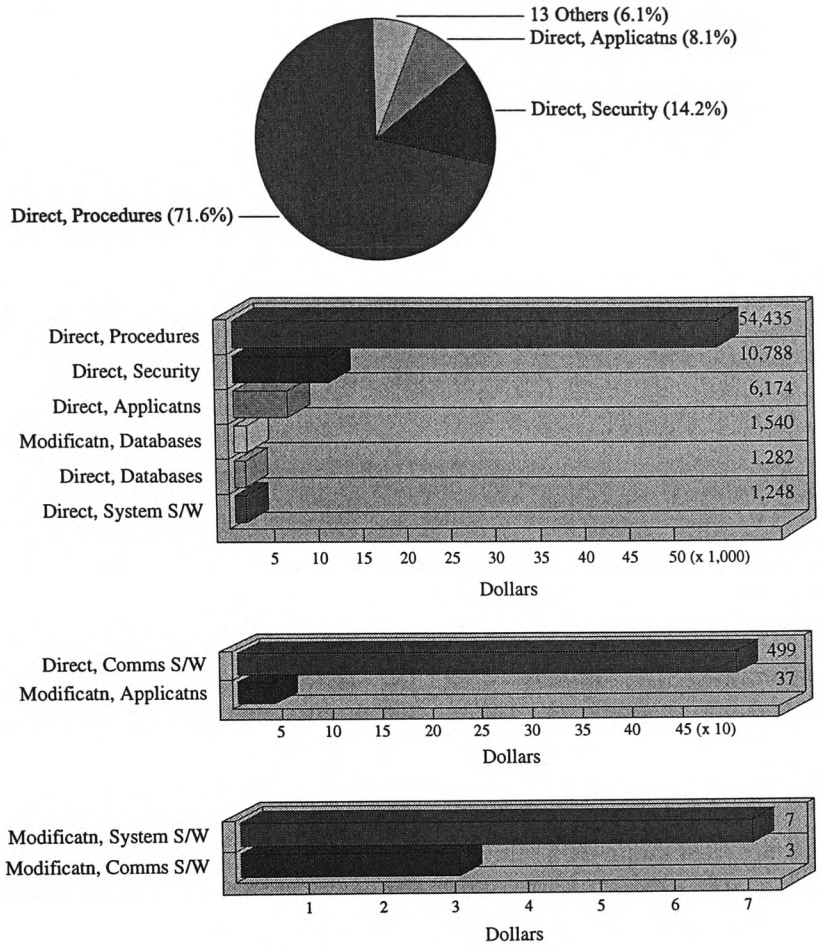


Figure 3 Data integrity loss—ALEs.

department should have the ability to turn remote access off at any time.

- Proper authentication and data encryption mechanisms (e.g. VPNs) should be put in place.
- The scope and techniques of data encryption should be expanded.
- Also, part of the stringent security measure is the need for UTVE to implement IDS technology to monitor the content of each connection. The recommendation is that a host-based IDS be run on the Web servers and a network-based IDS run at the border.
- Remote access use should be limited to only those needing it for business with technical and management safeguards.

In section "A case study on IDS ROI calculation modular approach", the ROI is calculated using risk management data and IDS implementation costs (for both the single support and MSSP support scheme) at UTVE for the different IDS deployment options discussed in section "Comparison of aggregate costs of different implementation schemes".

Calculation of ROI

Procedurally, once the Asset Valuations (AV and UEA), Exposure Factors (EF and EFs) have been calculated, the SLE and the ARO are then calculated.

In general, the ARO is computed based on the analysis of the annual frequency of threats. A distinction has to be made on the two types of ARO—Site-specific rate ARO and National ARO.

Based on the analysis and assumptions in each scenario, the computations for AV and UEA, EF and EFs and ARO are shown in Appendix 2.

For each threat, an ARO is derived by analyzing available national data. The derived ARO values developed from the national data are not as applicable as AROs developed with site-specific data. Site-specific data are defined as information gathered directly on or from the site itself such as those represented in Table 7. Historically recorded data of previous threat occurrences, which can generally be collected from any specific site are: maintenance logs, documentation on system operations and system failures, air-conditioning

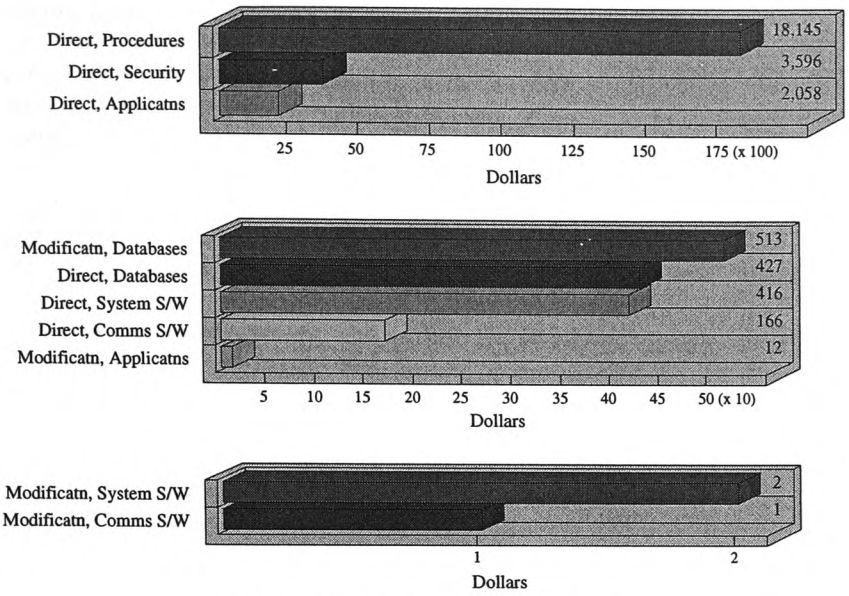


Figure 4 Data integrity loss—SLEs.

and power failure, and component mean-time between failure reports, etc. The following are the guidelines to follow when determining the ARO value for a given threat:

- When possible, the ARO value for the given threat is developed from site-specific/resident data. This requires gathering site resident data as needed to calculate the mean and standard deviation for any specific threat ARO. Examples of threats that are best represented by site-specific data are: air-conditioning failures, power outages, operator errors, user input errors, system crashes, and theft.
- When it is not practical to gather the site-specific data required to calculate the ARO value for a particular threat, the standard ARO value can be used.

An example of local site statistics on the rate of occurrence of known attacks is given in Table 7. These are actual numbers based on network attacks that NetSolve, Inc. manages (Ptacek and Newsham, 1999).

In Appendix 3, each variable and risk equation that was used in the ROI calculations for UTVE IDS deployment is itemized. A review of the Appendix shows how the traditional ROSI equation has been tied back to the ALE containing the CTM factor i.e.

$$\text{ROSI} = R - \text{ALE},$$

where the commonly accepted

$$\text{ALE} = (R - E) + T$$

is now replaced with

$$\text{ALE} = \text{ARO} \times \text{SLE},$$

and

$$\text{SLE} = \text{AV} \times \text{EF} \times \text{CTM}.$$

The support costs ($83,217/year for single support coverage and $44,217/year for MSSP support coverage) used to calculate these ROIs were taken

Table 7 Average attack occurrences per network

Attack	Per network attempts (April)	Per year attack attempts	Scenario
General Cmd.Exe	9492	113,904	1
Root.exe backdoor	1869	22,428	1
Ida overflow	105	1260	1
SSH attacks	2	2.0	2
DNS bind attacks	7	8.0	2
FTP attacks	3	4.0	2
Apache chunked	6	7.0	2
IOS HTTP unauthorised	3	36	3

from Table 3. The results of the calculations are shown in Appendix 4. The use of auto-response scheme produces by far better ROI in both NIDS and HIDS deployments. A conservative estimate of 50% reduction in ARO is facilitated by the utilization of auto-response. Also, the 25% reduction in both exposure factors (EF and EFs) should also be considered a conservative estimate in the IDS deployment with auto-response and prompt incident response scheme.

The benefits of a better IDS management are reflected in the reductions in the values of the variables in the highlighted cells under ARO, EF and EFs in Appendix 4. The overall effect is visible in the increase in the ROI values for the UTVE IDS deployment for both the single in-house support and MSSP support schemes.

Auto-response affects primary mitigation windows, which has a direct impact on partially reducing the Annual Rate of Occurrence (ARO). This is illustrated in the ROI Appendix 4, where a beneficial conservative reduction in ARO of 50% (highlighted in yellow (web version) in the "IDS w/Auto-Response" rows for each of the three scenarios) is attained. Incident response affects the secondary mitigation window, which impacts Exposure Factor (EF) and Secondary Exposure Factor (EFs), which in turn impacts the Cascading Threat Multiplier (CTM). This is also illustrated in the ROI in Appendix 4, where a beneficial conservative reduction in EF and EFs of 25%, respectively (highlighted in yellow (web version) in the "IDS w/ Auto-Response & Incident Response" rows for each of the three scenarios) is attained.

These reductions have positive effects on the IDS ROI. Once the aggregate annualized savings (ALE1−ALE2 or ALE1−ALE3) occurring from IDS deployment equals the support costs associated with the deployment a positive ROI should materialize. In the case of UTVE, the two ROIs (ROI1 and ROI2) for each support profile are as follows:

- single support with IDS using auto-response (ROI1) = −4%;
- single support with IDS using auto-response and incident response (ROI2) = 36%;
- MSSP support with IDS using auto-response (ROI1) = 81%; and
- MSSP support with IDS using auto-response and incident response (ROI2) = 155%.

These ROIs are based on the aggregate annualized savings from deploying and effectively managing the IDS technology and the resulting impact the IDS technology could reasonably have on the combined effect of the three compromise scenarios described above (Appendix 4).

In the final analysis, attainment of a better ROI depends on a good management practice especially the use of highly skilled engineers or technicians who have a sound understanding of the technology including the inherent strengths and weaknesses to manage the IDS technology. It is also a reasonable assumption that a single in-house engineer or technician would better support IDS deployment of one NIDS and two HIDSs. On the other hand, it will be ineffective to assume that one person can support this highly dynamic technology on a continual 24/7/365 basis with active auto-response and real-time incident response for every security event. Multi-shift internal support as well as Managed Security Service Provider (MSSP) support is the preferred ways of providing definitive 24/7/365 support and real-time incident response.

Conclusions

The importance of using intrusion detection as a means of risk management has been pointed out by several researchers. This work in ROI modeling for IDSs has benefited from the insightful analysis from real-world experiences demonstrated in the case study and draws from research in intrusion detection systems using knowledge gained from security risk management.

The contributions made by this paper are in the development and introduction of a new concept—Cascading Threat Multiplier (CTM) and the model framework used to accurately calculate the ROI for any acquired or deployed IDS technology.

To effectively analyze and calculate the IDS ROI, there is the need to have a sound understanding of the environment where the IDS is deployed including, at a minimum, the business practice, and network architecture and asset values. Equally, a good analysis of system vulnerabilities and associated threats should be addressed within the framework of a sound security policy and risk mitigation techniques.

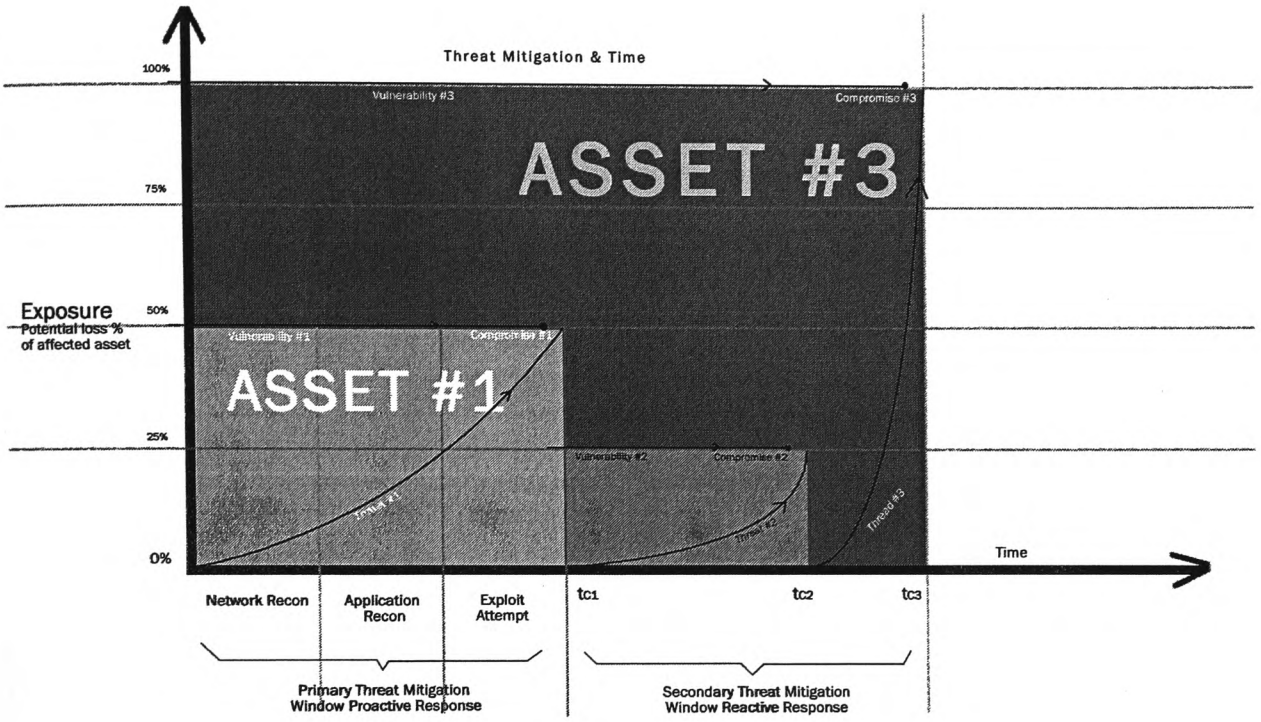
Finally, this paper has demonstrated that a positive IDS ROI is attainable with an effective deployment technique and optimal management approach.

Acknowledgements

I would like to acknowledge the following people who have contributed to this work in one manner or another: Professor Andrew Blyth, School

of Computing, University of Glamorgan, Wales, UK for his constructive suggestions; Professor Mukesh Singhal, Department of Computer Science, University of Kentucky, Kentucky, USA for his editorial assistance; Kevin Timm and David Kinn, Security Engineers at Netsolve, Inc., Austin, USA for providing some of the data in Table 7. To these people, I am extremely grateful.

Appendix 1. Threat mitigation window



Appendix 2. Calculations for asset valuations, exposure factors and annual rate of occurrence

Scenario	Descriptions of Compromised Asset (AV) and Underlying Exposed Assets (UEA)	Considerations in assigning estimates for Asset Valuations (AV & UEA), Exposure Factors (EF & EFs) and Annual Rate of Occurrence (ARO)	AV	EF	UEA	EFs	ARO
One	NT 4.0 Web server (AV); NT Domain (UEA)	Cost of lost productivity/revenue from downtime? Cost of compromised underlying data/assets? Cost of rebuilding web server? Potential cost of compromise of NT domain resources?	$2,000	75%	$20,000	75%	3
Two	UNIX-based Web server (AV); old internal dbase containing inventory data/pricing for customers and suppliers (UEA)	Cost of lost productivity and revenue from downtime? Cost of loss of trust or confidence of UTVE's online customers? Cost of compromised data and assets? Cost of rebuilding web server? Immediate cost of fulfilling current orders to satisfy customers?	$2,000	50%	$50,000	50%	2
Three	Router; Primary supplier UTVE's network	Cost of supply interruption from primary supplier? Cost of loss of trust or confidence of primary supplier? Potential cost of compromised data? Cost for UTVE to replace ACME as a supplier? Difference in credit terms for new supplier (compared to highly favorable terms that UTVE currently enjoys with ACME)? Difference in pricing between normal (new) supplier and ACME's pricing? Potential cost of litigation if ACME determines UTVE employee is at fault? Cost of UTVE compromise? Potential cost of liability for attacks directed at other non-partner networks?	$3,000	75%	$200,000	50%	1

Appendix 3. ROI variables and risk equations

Variable	Formula or expression
Asset Value	AV = hardware + comm. software + proprietary software + data
Exposure Factor	EF is the % estimation of the exposure of the initial compromised asset
Underlying Exposed Assets	UEA is the estimation of the $ value of the assets behind the compromised initial asset
Secondary Exposure Factor	EFs is the % estimation of the exposure of the UEAs
Cascading Threat Multiplier	CTM = 1 + ((UEA x EFs) / AV)
Single Loss Expectancy	SLE = EF x AV x CTM
Annual Rate of Occurrence	ARO is estimated number, based on available industry statistics or experience
Annual Loss Expectancy without IDS	ALE1 = SLE x ARO
Annual Loss Expectancy with IDS using auto-response	ALE2 = conservative 50% reduction of ARO when IDS is managed skillfully with auto-response
Annual Loss Expectancy with IDS using auto-response & incident response	ALE3 = conservative 25% reduction of EF & EFs when IDS is managed skillfully with auto-response and incident response
Annual Cost (T) of IDS Technology and Mgmt	T
Annual Recovery Cost (R) from Intrusions without IDS	R = ALE1
Annual Dollar Savings (E) gained by stopping intrusions with IDS	E = ALE1 - (ALE2 or ALE3)
Traditional Return on Security Investment (ROSI) equation	ROSI = R - ALE, where ALE = (R-E) + T
UTVE IDS ROI with auto-response	ROI1 = ALE1 - ((ALE1 - (ALE1 - ALE2)) + T)
UTVE IDS ROI with auto-response & incident response	ROI2 = ALE1 - ((ALE1 - (ALE1 - ALE3)) + T)

Appendix 4. IDS ROI for different management schemes

SCENARIO	ROI SCOPE	AV	EF	UEA	EFS	CTM	SLE	ARO	ALE_1 (no IDS)	ALE_2 (w/ AR)	ALE_3 (w/ AR & IR)	Single Support ROI $	Single Support ROI %	MSSP Support ROI $	MSSP Support ROI %
ONE	no IDS	$2,000	75%	$20,000	75%	8.5	$12,750	3	$38,250	N/A	N/A	N/A	N/A	N/A	N/A
	IDS w/ Auto Response	$2,000	75%	$20,000	75%	8.5	$12,750	1.5	$38,250	$19,125	N/A	-$64,092	-77%	-$25,092	-57%
	IDS w/ Auto Response & Incident Response	$2,000	56%	$20,000	56%	6.6	$7,453	1.5	$38,250	N/A	$11,180	-$56,147	-67%	-$17,147	-39%
TWO	no IDS	$20,000	50%	$50,000	50%	2.3	$22,500	2	$45,000	N/A	N/A	N/A	N/A	N/A	N/A
	IDS w/ Auto Response	$20,000	50%	$50,000	50%	2.3	$22,500	1	$45,000	$22,500	N/A	-$60,717	-73%	-$21,717	-49%
	IDS w/ Auto Response & Incident Reponse	$20,000	38%	$50,000	38%	1.9	$14,531	1	$45,000	N/A	$14,531	-$52,748	-63%	-$13,748	-31%
THREE	no IDS	$3,000	75%	$200,000	50%	34.3	$77,250	1	$77,250	N/A	N/A	N/A	N/A	N/A	N/A
	IDS w/ Auto Response	$3,000	75%	$200,000	50%	34.3	$77,250	.5	$77,250	$38,625	N/A	-$44,592	-54%	-$5,592	-13%
	IDS w/ Auto Response & Incident Response	$3,000	56%	$200,000	38%	26.0	$43,875	.5	$77,250	N/A	$21,938	-$27,905	-34%	$11,096	25%
WBS ROI with IDS Auto- Response (ROI_1)									$160,500	$80,250	N/A	-$2,967	-4%	$36,033	81%
WBS ROI with IDS Auto- Response & Realtime Incident Response (ROI_2)									$160,500	N/A	$47,648	$29,625	36%	$68,625	155%

References

Available from: http://www.nss.co.uk/Articles/IntrusionDetection.htm.

Available from: http://www.silicondefense.com/software/acbm/speed_of_snort_03_16_2001.pdf.

Iheagwara C, Blyth A. Evaluation of the performance of IDS systems in a switched and distributed environment. Comput Netw 2002;39:93–112.

Iheagwara C, Blyth A, Singhal M. A comparative experimental evaluation study of intrusion detection system performance in a gigabit environment. J Comput Secur January 2003;11(1).

Irvine C, Levin T. Toward a taxonomy and costing method for security metrics. In: Proceedings of the Annual Computer Security Applications Conference, Phoenix, AZ; December 1999.

Kevin T, Kinn D. CTM. Technical Report, Netsolve, Inc., Austin, USA; 2002.

Lee W, Fan W, Miller M, Stolfo S, Zadok E. Toward cost-sensitive modeling for intrusion detection and response. North Carolina State University; 1999.

Ptacek TH, Newsham TN. Insertion, evasion, and denial of service: eluding network intrusion detection. Secure Networks, Inc.; 1999.

Richards K. Network based intrusion detection: a review of technologies. Comput Secur 1999;18:671–82.

Charles Iheagwara is the Director of IT Security Services at Una Telecom, Inc. in Lanham, Maryland, USA. In this position, he oversees all IT security projects for private and government clients. He is also one of the Principal Investigators for corporate research, which are geared towards studies on enterprise-wide security solutions. These include translating IT security academic and laboratory research into business solutions.

He is also an External Research Fellow at the School of Computing, University of Glamorgan, Wales, where he specializes in Intrusion Detection Systems research and development. He has published numerous technical and scientific papers in the field in the refereed journals, conference proceedings and research newsletters.

He is also an Adjunct Professor of Computer Science in the School of Business Studies at Bowie State University, in Maryland, USA where he teaches graduate-level information assurance courses.

Appendix 9

"Cost Effective Management Frameworks for Intrusion Detection Systems" Journal of Computer Security, Volume 12, Number 5, 2004, pp. 777-798

Journal of Computer Security 12 (2004) 777–798
IOS Press

Cost effective management frameworks for intrusion detection systems

Charles Iheagwara [a], Andrew Blyth [b] and Mukesh Singhal [c]
[a] *Una Telecom, Inc., 4640 Forbes Boulevard, #200, Lanham, MD 20706, USA*
[b] *School of Computing, University of Glamorgan, Pontypridd, Wales, CF 37 1DL, UK*
[c] *Department of Computer Science, The University of Kentucky, 773 Anderson Hall, Lexington, KY 40506, USA*

This paper discusses the financial benefit of intrusion detection systems (IDS) deployment techniques and addresses the problems of bridging the gap between technical security solutions and the business need for it. This is an area of interest to both the research and the business community; most IDSes balance host and network monitoring, but the decision about how to adjust usage of each technique tends to be made in a rather ad-hoc way, or based upon effectiveness of detection only without regard to cost of technique. In practice, selections based on how well a strategy helps a company to perform are preferable and methodologies supporting a selection process of this type will assist an Information Technology officer to explain security mechanism selections more effectively to CEOs. In this context, the approach we propose could be applied when choosing one intrusion detection system over another based on which has a better or higher return on investment for the company.

Through a case study, we illustrate the benefits of a better IDS management that leads to a positive Return on Investment (ROI) for IDS deployment. We conceive strategies and approaches to support effective decision-making about which techniques are appropriate for the cost effective management of the IDS in a given environment. It is our intent that this research will serve as a foundation for the formal description of cost structures, analysis, and selection of effective implementation approaches to support the management of IDS deployments.

1. Introduction

Independent of implementation costs, the method in which security devices such as IDSes are managed can have a serious effect on the Return on Investment (ROI). Thus, a positive ROI for the IDS is dependent upon an organization's deployment strategy and how well the successful implementation and management of the technology helps the organization achieve the tactical and strategic objectives it has established.

However, given the high cost of IDS deployments especially when multiple deployments are involved, organizations must justify implementation expenses by proving that the IDS is a value added resource. One possible justification is to establish that the deployment of the IDS should lead to a reduction in the annual loss expectancy (ALE) and the return on security investment (ROSI).

One method for justifying IDS is by determining the value of the ALE using conventional cost/benefit (risk) assessment; the ALE represents the cost/benefit break-even point for risk mitigation measures. In other words, the organization could justify spending up to the dollar amount equivalent of the ALE per year to prevent the occurrence of loss or reduce the impact of a cyber attack for example. An alternative method for justifying IDS is to demonstrate the ability of the IDS to effectively detect and deter attacks in cost-effective quantifiable measures or to implement it as a standard due care measure. There are prior research studies [1–3] on this. Another option is to analyze the benefits of IDS by documenting the misuses of an organization's network

· Hence, for many organizations, investment decisions on IDS deployment will hinge on the ability to demonstrate a positive ROI and are not just motivated by the needs of security risk management. For the IDS to be cost-effective, it should cost no more than the expected level of loss from intrusions. This requires that the IDS *purchaser* consider the trade-off among cost factors [4,5], which at the minimum should include the cost of damage or compromised asset due to an intrusion, the cost of manual or automatic response to an intrusion, and the operational cost, which measures constraints on time and computing resources. For example, an intrusion where the response or mitigation cost is higher than the damage cost should usually not be acted upon beyond simple logging.

Therefore, implementation costs are very important and should be among the determinant factors for effective IDS management. Although in current IDS implementations, cost value propositions are rare due to the complexities of the networked environment in which they are deployed. Another reason for this is the fact that many organizations are not educated about the cost-benefits of security systems and for some, analyzing site-specific cost factors could be very challenging [6].

The challenge could be partly attributed to the difficulties in the assessment of costs related to computer security, in part because accurate metrics have been inherently unrealistic. Of those costs that can be measured, the largest in terms of monetary value typically involve theft of proprietary information or financial fraud. Others that are more difficult to quantify but have resulted in severe loss of use or productivity include viruses and malware, Web server denial-of-service attacks, abuse of access privileges, and equipment vandalism or outright theft. The challenge is also due to the fact that cost structures and cost management (and costing for that matter) of IT security devices have not been extensively studied; at least not very well documented in technical or scientific literature. The few available studies have been presented from different perspectives.

In the business arena, management costs are calculated through cost benefit analysis (CBA) models/equations with a high degree of accuracy. Here, the models incorporate the use of risk-adjusted cash flows in order to examine internal rate of return (IRR) and maximum net present value (NPV) figured as a percentage of information security expenditures. The basis for this is the observation that a simple return on investment (ROI) calculation that divides income by asset value is insufficient because

it is based on historical rather than future valuations as affected by breach incidents. A more elaborate discussion of CBA is given in Section 2.

The increase in the use of IDS products mandates formulation of appropriate frameworks for their cost-effective management. Such frameworks among others could be used to translate existing cost models into technical solutions as implementation cost structures. This could be realized by first developing the cost metrics and then integrating them with existing theoretical cost models developed in previous studies within the contexts of the frameworks we propose in this research.

The insights gained from previous research studies that describe proven techniques to implement the technologies could be helpful in understanding effective management techniques for IDS deployments. Research in the area of cost modeling for network intrusion detection systems typically follow a risk analysis procedure to select sensitive data/assets and create a cost matrix for each intrusion.

Wei et al. [7] propose a cost-benefit analysis methodology and build a cost model that can be used to quantitatively and qualitatively calculate the cost of detecting and responding to an intrusion.

Lee [4] and Stolfo's [5] studies the problem of building cost-sensitive intrusion detection models and define cost models to formulate the total expected cost of IDS and examine the major cost factors associated with IDS, which include development cost, operational cost, damage cost due to successful intrusions, and the cost of manual and automated response to intrusions. The cost components related to intrusion detection are:

- Damage cost;
- Operation cost; and
- Response cost.

Combining the above cost components Lee [4] proposes a cost matrix for a risk analysis calculation:

$$\text{Cost\_total}(e) = \sum_{i=1}^{N} (\text{CCost} + \text{OperationCost}(e)). \tag{1}$$

In the above formula, Cost total(e) is the total cost for some event e, N is the event number, and CCost is the consequential cost of the prediction by the network intrusion detection system for the intrusion event e, which is determined by the damage cost and response cost. The Damage cost (DamageCost) represents the maximum amount of damage to an attack target when the intrusion detection system and other protective measures are either unavailable or ineffective. The Response cost (ResponseCost) is the cost of responding to the intrusion, which includes taking some action to stop the intrusion and reduce the damage. These actions or countermeasures should be defined during the risk analysis process according to specific threats. Operation cost (OperationCost) is the cost of processing the stream of events being monitored by an intrusion detection system and analyzing the activities using intrusion detection models.

Table 1
Security cost examples

Security service	Service area	Mechanism	Cost measure
Data confidentiality	NC	Link layer 40-bit DES	Processor clocks per byte
Message non-repudiation	ES	Remote non-repudiation service	$2n$ bytes per message network bandwidth, plus c clocks per byte
Intrusion detection	TS	Experimental system	N Mbytes per second of overall bandwidth, plus m instructions per second, plus b bytes per second storage

Thus, Lee's [4] major contribution to IDS cost models is that he proposed a cost matrix that combines the different cost features defined above for a risk analysis calculation.

For intrusion detection, Irvine [8] defines auditing of network control functions in intermediate nodes, and rule-based network intrusion systems in the total subnet as the mechanisms. Irvine also discusses the costs of those security services and mechanisms (Table 1).

In Irvin's proposition, security services include data confidentiality, integrity, traffic flow confidentiality, authenticity, non-repudiation, availability, audit and intrusion detection, and boundary control. For the three service areas delineated for security service analysis, a client or server system is an example of ES, routers and switches are the examples of IN, and NC indicates the wires that connect systems and nodes. Additionally, Irvin defines Total Subnet (TS) as a service area that can't be assigned exclusively to IN, NC or ES and defines at least one security mechanism for each security service and service area. For example, to protect data confidentiality, he defines operating system and cryptographic credentials as the security mechanism in the ES and IN's. Irvine also defines auditing of network control functions in IN and rule-based network intrusion systems in TS as the mechanisms.

The above propositions are difficult to realize because the units of the cost measure are impracticable to use. Equally, the lack of a quantitative and qualitative cost-benefit analysis and cost benefit tradeoff criteria for the computer security services complicates the application of the proposition.

In another cost model [9], five different prediction cases are identified as False Negative (FN), True Positive (TP), False Positive (FP), True Negative (TN) and Misclassified Hit.

False Negative (FN) is the cost of not detecting an attack. FN is incurred either by a system that does not install an intrusion detection system, or one in which the intrusion detection system does not function properly and mistakenly ignores an attack. This means that the attack will succeed and the target resource will be damaged. The FN cost is therefore defined as the damage cost of the attack. True Positive (TP) occurs in the event of a correctly classified attack, and involves the

cost of detecting the attack and responding to it. This is represented by the formula "Progress × DamageCost", where Progress is the percent of the attack's progress.

False Positive (FP) occurs when an event is incorrectly classified as an attack. True Negative (TN) cost is always 0, as it is incurred when a network intrusion detection system correctly decides that an event is normal. Misclassified Hit cost is incurred when the wrong type of attack is identified. If the response cost is less than the damage cost, a response action will be taken to stop the attack. Since the action is not useful for the actual attack, some damage cost occurs due to the progression of the true attack.

The above cost model [9] may be impracticable to use and it is not clear how to account the cost for management, maintenance, etc.

In contrast to the above, our contribution in this study is to use reverse engineering technique to formulate appropriate cost-effective management frameworks for IDS implementations. Using the knowledge and experiences gained in the implementation of IDSes, we demonstrate how different management techniques affect the return on investment and will then craft the frameworks around these experiences to improve operational and implementation costs. The frameworks we propose are effective in assessing network intrusion detection systems. They can be used to periodically review the effectiveness of planned and implemented IDSes to determine if they are doing what they are supposed to do, rather than add more cost than the anticipated benefit.

The rest of the paper is organized as follows. In Section 2, we review the state-of-the-art of cost benefit analysis techniques that have been proposed by other researchers. Implementation approaches are discussed in Section 3 and management and costs structures are presented in Section 4. In Section 5 we present a case study that explores the effects of different implementation schemes on the return on investments. We then propose effective management frameworks in Section 6 and conclude our discussion in Section 7.

2. The state-of-the-art of cost benefit analysis techniques

One of the most important problems facing information assurance is coming up with a method that accurately calculates the costs associated with lost. This in part is because accurate metrics have been inherently unrealistic. Of those costs that can be measured, the largest in terms of monetary value typically involve theft of proprietary information or financial fraud. Others that are more difficult to quantify but have resulted in severe loss of use or productivity include viruses and malware, Web server denial-of-service attacks, abuse of access privileges, and equipment vandalism or outright theft. Results of surveys of organizations provide estimates as to breach incidents, security expenditures, malicious code, and so on, with numbers continuing to reflect dramatic growth each year. However, lacking any way to translate such statistics into expenditures and losses per organization, per computer, or per user, the

true impact of these figures remains uncertain. An alternative method has been to use the Annual Loss Expectancy (ALE) to estimate risks and hence project potential loses that could result from the risks materializing.

The ALE, a quantitative method for performing risk analysis has been used as one of the earliest estimators in the computer industry was. The ALE is used to calculate risk estimates by multiplying the estimated frequency of occurrence of attacks by the possible loss amount for each data file, and then summing these results. The method has been criticized because of the "lack of empirical data on frequency of occurrence of impacts and the related consequences" thus producing an interpretation of "results as having more precision than they actually had" [10]. Nevertheless, the ALE figures may still provide some useful information. As a result, information technology companies are now resorting to using the established Cost-Benefit Analysis (CBA) method.

The CBA, which has become the most popular metrics, is applied to the assessment of computer-related risks. CBA is well established in microeconomic and management accounting theory, and can be used to determine estimated levels of expenditures appropriate to the values of assets requiring protection. Hazlewood [11] contends that it is particularly convincing since "most managers and directors know little about computers and computer security, but they do understand risk and cost-benefit analysis". The National Institutes of Health (NIH) has a useful guidance document for preparing CBAs as required by the US. Federal government to support IT management decisions [12]. Although the NIH document does not specifically pertain to security, many of the IT topics and examples discussed are highly relevant, so it is worth a close look.

CBA concepts are distinctively multiple depending on the manner and environment of their application. In IT fields, CBA models are increasingly becoming important in cost estimations and have been effective in assessing network intrusion detection systems. The process involves first performing a risk analysis that produces a cost matrix for the assets under attack, and then independently calculating damage, response, and operation costs for those assets. Resources to counter the attack can be classified as low, medium, or high, in terms of price, and weighted by amounts of use where appropriate, to obtain total expenditures. Probabilistic models also include false negative and false positive costs, since these may have an impact on losses.

An example of early CBA use in computer security is in the I-CAMP (Incident Cost Analysis Modeling Project) model developed by the Big Ten Universities during the 1990s. Factored together are the time, wages, overhead, and direct costs related to the resolution of individual security incidents. Person-hours are logged, typically for incident investigation, system administration, and recovery efforts and then salary-weighted sums (including benefits) are computed. Necessary direct expenditures (such as for replacement hardware, software, and analysis tools) are also added. The I-CAMP model is appropriate for situations where the related usage losses are considered to be modest or ignored entirely.

There are other costs that may be incurred with security protection mechanisms even when provided for free (as in the case of automatically downloaded software patches). Researchers on a DARPA-funded project [13] developed "a mathematical model of the potential costs involved in patching and not patching at a given time". They observed that the risk of loss of functionality from applying a bad patch decreases in time, while the risk of loss due to penetration while the patch is not applied increases with time. They hypothesized that the optimal time to apply the patch is when these curves cross, and developed a mathematical model (similar to the weighted ROI) that took into account various cost and probability factors. Using data collected from a study involving 136 patches, they were able to determine that at 10 and 30 days following a patch release, application is optimal. Of course, these intervals rely on some folks applying the (potentially bad or even bogus) patches sooner and reporting the defects they experienced – if everyone waits for the patches to be fixed, the time would be shifted forward, thus increasing early penetration risks.

There are also potential misuses of the CBA. Among these is in the application of the CBA to public-key cryptography in order to derive appropriate key sizes and expirations.

Silverman [14] asserts that a financial model, rather than a purely computational one, should be used to assess cryptographic vulnerabilities. He says "it makes no sense for an adversary to spend (say) $10 million breaking a key if recovering the key will only net (say) $10 thousand".

The CBA has also not been without problems in terms of use and acceptance. One of the major impediments in the use of CBA is the complexity of the equations. This has been a problem in the business arena where CBA equations are considered more complex. Here, the models incorporate the use of risk-adjusted cash flows in order to examine internal rate of return (IRR) and maximum net present value (NPV) figured as a percentage of information security expenditures. Gordon and Loeb [15] explain that a simple return on investment (ROI) calculation that divides income by asset value is insufficient because it is based on historical rather than future valuations as affected by breach incidents. They use weighted annual expected loss estimates derived by multiplying the dollar value associated with potential breaches by the probability of occurrence for each breach. But they note that even the IRR and NPV metrics may be deficient because these compare the actual cost savings from the security investment to the anticipated cost savings, which "is difficult because the benefits of specific investments aren't easily separated from other activities within a company. This is particularly relevant to security investments, the more successful the project, the less likely you are to see breaches".

All of this presents the opportunity to broaden the scope and dept of cost-benefit analysis using a multi-faceted approach and also to address business process concerns in the hope that empiricism can shift the balance in favor of the consumers of computer security products and services. In the process, those "add-ons" and providers that do not demonstrably improve the security cost bottom line will be exposed and dispensed with. And as a necessity new tools and metrics that enable

risk and cost-benefit assessments will be developed and proliferated. Only through such independent quantification can we hope to get a true handle on the financial ramifications of security problems so that we might best direct our efforts toward resolving them.

With the above in mind, it needs to be pointed out that recent studies point to the direction of crafting new CBA techniques through interactive or adaptive techniques. Shawn [6] uses a cost-benefit analysis method called SAEM to compare alternative security designs in a financial and accounting information system. The goal is to help information-system stakeholders decide whether their security investment is consistent with the expected risks.

3. Implementation approaches

Although there are many different approaches to intrusion detection, we believe that all of these variations can be categorized into two basic approaches, reactive and proactive. We will provide a context-based analysis on how each approach affects management cost in Section 6.3.

3.1. Reactive approach

We define a reactive approach as one in which response is done once personnel have been enlisted. Reactive approaches generally rely on techniques, such as cryptographic checksums or audit trail analysis mechanisms. A good example of a widely used UNIX IDS utility is Tripwire. Tripwire allows the files of a UNIX operating system to be cryptographically sealed for later review and comparison. If a file is modified, the checksum won't match, and an intrusion can be assumed. Several other passive IDS tools are available either as commercial products or as freeware/shareware. In almost every case, the tools provide a "post-mortem" of a security event or action. Since the tools don't monitor data transactions or other real-time events, they don't provide a means of preventing unauthorized intrusions. Instead, they provide a means to quickly respond to a security compromise, and in some cases, act as a deterrent to would-be system intruders.

3.2. Proactive approach

We define a proactive approach as one in which response is automated by the system. The proactive approach is based on active monitoring and analysis. Tools and utilities, using active techniques, monitor the actual data traffic, keystrokes, or other actions, and compare them against some predefined set of thresholds or rules. If a threshold or rule is exceeded, an alarm is activated. The key concept in active monitoring systems is that of real-time data collection, analysis, and alarms.

The distributed intrusion detection system (DIDS) is an example of proactive network-based IDS. DIDS is UNIX-based IDS that includes both agent software running on network hosts and a central security management console (the DIDS Director), where data is fused and alarms are generated in a graphical user interface. Developed by students at UC Davis, DIDS captures TCP/IP data traffic, in real time, and compares the collected traffic to stored "hacker profiles". If the system detects a condition that appears to indicate an unauthorized intrusion, an alarm is generated at a console, much like a traditional network management system.

It should be noted that just because an IDS captures raw network traffic, it still might not provide active IDS capabilities. Many times, network-based IDS will capture packets or raw network traffic, store it to a file, and review it at a later date or time. In a recent analysis performed by Secure Networks, Inc. of several IDS products, most of the current product offerings were found to be based on passive IDS techniques.

Another approach to active intrusion detection is based on monitoring specific characteristics at the host operating system level, such as CPU utilization, memory utilization, input/output rates, etc. By creating a baseline (over time) of a system or collection of systems, these parameters can be monitored and measured to identify potential anomalous behavior. Since this data can be collected and analyzed in real time, it can be considered a proactive form of intrusion detection.

4. Cost and management structures

In order to prepare for the next section, we present a cost and management structure for IDS implementation in Section 4.1. Using a holistic approach, we analyze the cost aggregate for the different implementation schemes in Section 4.2.

4.1. Implementation cost

The associated cost of host-based intrusion detection systems (HIDS) deployments can vary depending on vendor and software versions. A good baseline is that agents can cost between $500 and $2000 each and consoles may cost in the $3000–$5000 range [16]. This does not always include OS, hardware or maintenance costs. Network intrusion detection systems can be deployed as stand-alone hosts with a possible management interface or distributed sensors and management console. Generally speaking, in the last couple of years commercially available sensors run in the $5000–$20000 area [16] depending on vendor, bandwidth and functional capabilities. Management consoles can be included free as part of the cost, or sold separately and can cost several thousand dollars depending on the vendor. This does not necessarily include hardware or back-end databases.

The total cost of implementing an IDS-based security solution depends on purchasing costs combined with the costs for managing the technology. Giving IDS

Table 2
Cost of individual components [17]

Expense	Value ($)
Network IDS	$10 000
Host IDS	$1000
Management station – NIDS and HIDS	$5000 (may not apply for all products)
Maintenance	15% of the cost of NIDS and/or HIDS
MSSP network IDS management per year	$24 000 ($2K per month)
MSSP host IDS management per year	$6000 ($500 per agent per month)
Engineer cost	$75 000 ($60 000 salary plus $15K benefits and admin)
Group manager cost	$100 000 ($80 000 salary plus $20K benefits and admin)

Table 3
Cost structures [17]

	Single support	24 × 7 × 365 Multi-shift support	MSSP support
Technology cost	$24 650	$24 650	$24 650
Management cost	$225 000	$1 425 000	$108 000
Total cost	$249 650	$1 449 650	$132 650
Average cost per year	$83 217	$483 217	$44 217
Average cost per device per year	$27 739	$161 072	$14 739

management duties to a person not skilled in IDS technology is a poor idea. Some standard implementation and management methods common to IDS deployments include using a Managed Security Services Provider (MSSP), utilizing a single in-house employee or technician, or enabling 24 × 7 × 365 multi-shift coverage in-house with a skilled technical staff. Of course the size of the organization and its' associated IT budget (or lack thereof) factor in to how the IDS technology will be deployed and managed. Tables 2 and 3 represent the generalized cost structure that we will use for our discussion and case study.

4.2. *Comparative analysis of aggregate costs for different implementation schemes*

An analysis of the aggregate costs for three different IDS deployments can be made based on the generalized cost structure in Tables 2 and 3. Tables 4 and 5 represent implementation (purchase) costs combined with life cycle management costs over a three-year period. The three scenarios include management by a single skilled in-house technician, management in which there are five shifts of skilled technicians providing 24 × 7 × 365 coverage, and management provided by an MSSP. It is very important to understand that full-service MSSPs will provide 24 × 7 × 365 coverage just like the multi-shift internal coverage provides. For completeness, we will review two different IDS deployments (one small and one medium) and consider the cost structure of implementing and managing them.

Table 4
Implementation and management cost of one network IDS and two host IDS [17]

	Single support	24 × 7 × 365 Multi-shift support	MSSP support
Technology cost	$24 650	$24 650	$24 650
Management cost	$225 000	$1 425 000	$108 000
Total cost	$249 650	$1 449 650	$132 650
Average cost per year	$83 217	$483 217	$44 217
Average cost per device per year	$27 739	$161 072	$14 739

Table 5
Implementation and management cost of 15 network IDS and 15 host IDS [17]

	Single support	24 × 7 × 365 Multi-shift support	MSSP support
Technology cost	N/A	$268 250	$268 250
Management cost	N/A	$1 425 000	$1 350 000
Total cost	N/A	$1 693 000	$1 618 250
Average cost per year	N/A	$564 417	$539 417
Average cost per device per year	N/A	$18 814	$17 981

From the numbers it is evident that in smaller IDS deployments the value proposition of MSSP support is very strong relative to internal 24 × 7 × 365 multi-shift support. In larger IDS deployments, the cost differential between internal (highly skilled) multi-shift coverage and MSSP coverage diminishes due to economies of scale on the internal multi-shift coverage side. Single support coverage is not a realistic option to consider when contemplating a deployment of 30 security devices. Also, this cost model does not take into account proprietary tools development necessary to manage several different types of technology (if that were the case) effectively.

5. A case study on cost effective management approach

In this section we will use a hypothetical case study [16] to demonstrate the efficacy of the different management approaches. To do this we shall derive a value for the return on investment of each management method. The results will then be used to articulate a management framework.

5.1. Framework for risk analysis and ROI computation

Studies on suitable management approaches that maximize the IDS ROI are not clearly established. Therefore, the use of this case study approach will permit in-depth exploration of the benefits of illustrating ROI analysis in order to determine the management technique that maximizes the IDS deployment. From the case study, we hope to glean some general concepts about intrusion detection system ROI and

determine the most effective management approach that will maximize the return on investment. By developing the examples, we also hope to develop a possible method of reasoning about IDS cost effective management approaches more generally.

The case study will be presented in the context of the risk assessment and ROI studies given in Sections 5.4 and 5.5. In order to prepare for the studies, we set up a hypothetical company called ABC, Inc. and through the case study present the threat and incidence scenarios needed to calculate ROI – which is the indicator for effective implementation and lifecycle management of the IDS deployments.

Table 6
ROI variables and risk equations

Variable	Formula or expression
Asset Value (AV)	AV = hardware + comm. software + proprietary software + data
Exposure Factor (EF)	EF is the % estimation of the exposure of the initial compromised asset
Underlying Exposed Assets (UEA)	UEA is the estimation of the $ value of the assets behind the compromised initial asset
Secondary Exposure Factor (EFs)	EFs is the % estimation of the exposure of the UEAs
Cascading Threat Multiplier (CTM)	CTM = 1 + ((UEA × EFs)/AV)
Single Loss Expectancy (SLE)	SLE = EF × AV × CTM
Annual Rate of Occurrence (ARO)	ARO is estimated number, based on available industry statistics or experience
Annual Loss Expectancy Without IDS (ALE1)	ALE1 = SLE × ARO
Annual Loss Expectancy with IDS using auto-response (ALE2)	ALE2 = conservative 50% reduction of ARO when IDS is managed skillfully with auto-response
Annual Loss Expectancy with IDS using auto-response and incident response (ALE3)	ALE3 = conservative 25% reduction of EF and EFS when IDS is managed skillfully with auto-response and incident response
Annual Cost (T) of IDS Technology and Mgmt	T
Annual Recovery Cost ('R) from Intrusions without IDS	R = ALE1
Annual Dollar Savings (E) gained by stopping intrusions with IDS	E = ALE1 − (ALE2 or ALE3)
Traditional Return on Security Investment (ROSI) equation	ROSI = R − ALE, where ALE = (R − E) + T
ABC, Inc. ROI of IDS with auto-response (ROI1)	ROI1 = ALE1 − (((ALE1 − (ALE1 − ALE2)) + T)
ABC, Inc. ROI of IDS with auto-response and incident response (ROI2)	ROI2 = ALE1 − (((ALE1 − (ALE1 − ALE3)) + T)

5.2. Risk assessment

A risk assessment (analysis) study [16] was conducted to quantify the loss associated with the occurrence of an incidence or a threat at ABC, Inc. In the analytical approach leading up to the calculation for ROI, commonly accepted formulas and definitions (Table 6) are used to calculate Asset Valuations (AV and UEA), Exposure Factors (EF and EFS), the single loss expectancy (SLE) and the Annual Rate of Occurrence (ARO). To fully explore the risk factors, three different scenarios of possible asset compromises were considered.

Procedurally, once the Asset Valuations (AV and UEA) and Exposure Factors (EF and EFS) have been calculated, the single loss expectancy (SLE) and the Annual Rate of Occurrence (ARO) are then computed. The Annual Rate of Occurrence (ARO) is computed based on the analysis of the annual frequency of threats and the computations for Asset Valuations (AV and UEA), and Exposure Factors (EF and EFS) and (ARO). The results of these calculations for our case study are shown in Table 7.

One of the results of the study is the recommendation to implement IDS technology to complement other security devices as a counter measure to future attacks. We now incorporate into our case study the different IDS implementation schemes described in Tables 4 and 5 in Section 4 in order to delineate the effect of each implementation scheme on the return on investment (ROI). The ROI will be the ultimate gauge of the effectiveness of the IDS management approach.

Consequently, in Section 5.3 we calculate the ROI using the data derived from the risk assessment study and IDS implementation management costs (for both single and MSSP support schemes) discussed in Section 4.

5.3. Return on investment

The formulas used for the ROI calculations are shown in Table 6. The support costs ($83 217/year for single support coverage and $44 217/year for MSSP support coverage) taken from the Table 4 are used in the ROI calculations. The results of the ROI calculation for the different IDS implementation are shown in Table 8.

5.4. Analysis of results

Auto-response affects primary mitigation windows, which has a direct impact on partially reducing the Annual Rate of Occurrence (ARO). This is illustrated [16] in the ROI Table 8 above, where a beneficial conservative reduction in ARO of 50% (highlighted in yellow in the "IDS w/Auto-Response" rows for each of the three scenarios) is attained. Incident response affects the secondary mitigation window, which impacts exposure factor (EF) and secondary exposure factor (EFs), which in turn impacts the Cascading Threat Multiplier (CTM). This is also illustrated in the ROI Table 8 above, where a beneficial conservative reduction in EF and EFS of

Table 7

Calculations for asset valuations, exposure factors and annual rate of occurrence

Scenario	Descriptions of Compromised Asset (AV) and Underlying Exposed Assets (UEA)	Considerations in assigning estimates for Asset Valuations (AV and UEA), Exposure Factors (EF and EFS) and Annual Rate of Occurrence (ARO)	AV	EF	UEA	EFS	ARO
One	ABC, Inc. NT 4.0 Web server (AV); ABC, Inc. NT Domain (UEA)	Cost of lost productivity and revenue from downtime? Cost of compromised underlying data and assets? Cost of rebuilding web server? Potential cost of compromise of NT domain resources?	$2000	75%	$20 000	75%	3
Two	ABC, Inc. UNIX-based Web server (AV); old internal ABC, Inc. database containing inventory data and pricing for customers and suppliers (UEA)	Cost of lost productivity and revenue from downtime? Cost of loss of trust or confidence of ABC, Inc.'s online customers? Cost of compromised data and assets? Cost of rebuilding web server? Immediate cost of fulfilling current orders with whatever it takes to satisfy customers?	$2000	50%	$50 000	50%	2
Three	ABC, Inc. router; Primary supplier ABC, Inc.'s network	Cost of supply interruption from primary supplier? Cost of loss of trust or confidence of primary supplier? Potential cost of compromised data? Cost for ABC, Inc. to replace ACME as a supplier? Difference in credit terms for new supplier (compared to highly favorable terms that ABC, Inc. currently enjoys with ACME)? Difference in pricing between normal (new) supplier and ACME's pricing? Potential cost of litigation if ACME determines ABC, INC. employee is at fault? Cost of ABC, INC. compromise? Potential cost of liability for attacks directed at other non-partner networks?	$3000	75%	$200 000	50%	1

Table 8
IDS ROI

Scenario	RQI scope	AV	EF	UEA	EFS	CTM	SLE	ARO	ALE1 (no IDS)	ALE2 (w/AR)	ALE3 (w/AR& IR)	Single support ROI		N5SP support ROI	
												$	%	$	%
One	no IDS	$2000	75%	$20 000	75%	8.5	$12 750	3	$38 250	N/A	N/A	N/A	N/A	N/A	N/A
	IDS w/Auto-Response	$2000	75%	$20 000	75%	8.5	$12 750	1.5	$38 250	$19 125	N/A	–$64 092	–77%	–$25 092	–57%
	IDS w/Auto-Response and Incident Response	$2000	56%	$20 000	56%	6.6	$7453	1.5	$38 250	N/A	$11 180	–$56 147	–67%	–$17 147	–39%
Two	no IDS	$20 000	50%	$50 000	50%	2.3	$22 500	2	$45 000	N/A	N/A	N/A	N/A	N/A	N/A
	IDS w/Auto-Response	$20 000	50%	$50 000	50%	2.3	$22 500	1	$45 000	$22 500	N/A	–$60 717	–73%	–$21 717	–49%
	IDS w/Auto-Response and Incident Response	$20 000	38%	$50 000	38%	1.9	$14 531	1	$45 000	N/A	$14 531	–$52 748	–63%	–$13 748	–31%
Three	no IDS	$3000	75%	$200 000	50%	34.3	$77 250	1	$77 250	N/A	N/A	N/A	N/A	N/A	N/A
	IDS w/Auto-Response	$3000	75%	$200 000	50%	34.3	$77 250	0.5	$77 250	$38 625	N/A	–$44 592	–54%	–$5592	–13%
	IDS w/Auto-Response and Incident Response	$3000	56%	$200 000	38%	26.0	$ 43 875	0.5	$77 250	N/A	$21 235	–$27 905	–34%	$11 096	25%
WBS ROI with IDS Auto-Response (ROI1)									$160 500	$80 250	N/A	–$2967	–4%	$36 033	81%
WBS ROI with IDS Auto-Response & Realtime Incident Response (ROI2)									$160 500	N/A	$47 648	$29 635	36%	$68 635	155%

25% respectively (highlighted in yellow in the "IDS w/Auto-Response and Incident Response" rows for each of the three scenarios) is attained.

These reductions have positive effects on the ROI of IDS. Once the aggregate annualized savings (ALE1 − ALE2 or ALE1 − ALE3) occurring from IDS deployment equals the support costs associated to the deployment, a positive ROI should materialize. In the case of ABC, Inc., the two ROIs (ROI1 and ROI2) for each support profile are as follows:

- Single support with IDS using auto-response (ROI1) = −4%;
- Single support with IDS using auto-response and incident response (ROI2) = 36%;
- MSSP support with IDS using auto-response (ROI1) = 81%; and
- MSSP support with IDS using auto-response and incident response (ROI2) = 155%.

These ROIs are based on the aggregate annualized savings from deploying and effectively managing the IDS technology and the resulting impact the IDS technology could reasonably have on the combined effect of the three compromise scenarios described above (see Table 7).

6. Propositions for cost effective management frameworks

The case study presented in the preceding section provides the insight needed to articulate the frameworks for a cost effective IDS management approach. From these, we propose the following management frameworks for the cost effective management of IDS deployments:

- Developing a composite metrics for cost estimates;
- Using local environmental factors to optimizing product selection;
- Implementation with the proactive and auto response mechanism; and
- Adopting a cost effective staffing and support structure.

6.1. Developing a composite metrics for cost estimates

Developing a composite metrics from known implementation cost items will in the longer run help establish a somewhat accurate budget for IDS management. Apart from the functional requirements, the IDS must also satisfy a number of economical requirements, in particular, cost. The following cost categories are integrated into the costs structure of IDS management [18]:

- Cost of the IDS product.
- Cost of additional computer resources needed.
- Cost of administration.

In addition, the costs components (technology, management, maintenance) described in Section 4.1 should be factored into the costs of acquiring additional computer resources and administration. Together all of this will be used to create an IDS cost metrics.

The importance of these cannot be under estimated. This will then become a reference for large, medium and small size enterprises, or indeed anyone trying to implement an IDS security solution.

6.2. *Using local environmental factors to optimizing product selection*

Iheagwara et al. [3] demonstrate that IDS performance is greatly influenced by IDS product selection for a given environment. The study investigates the relationship between different IDS products performance in varied network and traffic stream conditions; and also provides a side-by-side comparison of two different technologies for intrusion detection. One being older (Megabit IDS) and the other (Gigabit IDS) representing evolutions from pure megabit IDS to gigabit IDS based on the extension of recurrent characteristics of ID system to new technologies.

Given that Gigabit requirements will increasingly become mandatory especially for carrier networks that are associated with problems of data management and information overload, the results are significant because the data on which the techniques are evaluated represent a significant corpus of empirically obtained data. This can be used to simultaneously measure and evaluate the probability of detection of a given intrusion detection technique against that of another technique in order to compare the correct detection rates. Detection rates are among the many criteria used in selecting the most feasible IDS product.

Iheagwara et al. [3] demonstrate that the IDS performance in large-scale infrastructures is directly related to traffic and environmental characteristics.

Operationally, the requirements of an enterprise network that is deploying a few devices locally to watch over a class-C network is going to be different from that of a multi-national corporation that is deploying hundreds of devices. The requirements should tie known performance values with the IDS product selection. In this regard, a clear delineation of the traffic load in order to estimate the type or number of a particular IDS product that will match expected performance level should precede the management approach.

In a different study, Iheagwara et al. [2] provide justification that an effective ID system can be achieved by using a best effort delivery/deployment approach that integrates the monitoring and deployment techniques to maximize the benefits of the ID system. The effectiveness of the IDS is closely linked to various factors including network topology, deployment techniques, and network throughput, bandwidth and network traffic conditions.

The conclusions drawn from the studies are that IDS product selection should be based on local factors. Comparatively, cost and other techno-economic factors should determine the product type and the implementation technology.

6.3. Implementation with the proactive and auto response mechanism

The concepts of proactive and reactive management techniques have been explained in Section 2 above. The sequence of events for each technique is explained in Table 9.

By examining the Annual Loss Expectancy (ALE = ARO * SLE, where SLE = Exposure Factor * Asset Value * Cascading Threat Multiplier) we can determine which variables are affected by each of these two management methods. In a reactive design, where personnel must be engaged to respond to each event, the exposure factors (primary [EF] and secondary [EFs]) will be affected. In a proactive design there will be similar benefits to the exposure factors (re: a reduction) and, in addition, the Annual Rate of Occurrence (ARO) will be influenced in a beneficial way as well. To demonstrate the impact of threat vs. time we will use the concept of primary and secondary mitigation windows. In Fig. 1, the primary mitigation window affects ARO while the secondary mitigation window affects Exposure Factor and Cascading Threat Multiplier [16]. CTM factors in the importance of other critical assets tied (re: networked) to the specific asset being analyzed in the ALE calculation. An effective way of impacting ARO is through automated response.

Auto-response can take many forms. On host-based IDS this is sometimes called shielding, where a specific process is terminated. Network-based IDS generally employs TCP resets or shunning. TCP resets effectively kills one specific session based on suspicious activity, but it still allows other activity from that same IP. Shunning, on the other hand, changes firewall rules or router access lists and effectively denies all traffic from that host for a specific period of time. In essence, shielding will protect a single host from one process, resets will protect a host from a specific session, and shunning will protect the entire network from a specific host for a pre-determined amount of time.

The accuracy of automated response can vary tremendously. This is dependent on the skill level of the engineers managing the devices. If the engineers are moderately skilled then auto-response will not be very effective, which may adversely affect the ROI of the IDS deployment. This adverse effect may manifest itself in the form of a loss of productivity from network-related problems due to improperly implemented auto-response, as well as the additional fallout related to a false sense of security throughout the company. It is assumed that a moderately skilled engineer is one who has gone through a formal process of training on intrusion detection systems operation and must have managed or operated the device for at least twelve (12)

Table 9
Proactive and reactive management methods

Method	System actions	Personnel actions	Follow up information
Reactive	Log → Alert →	Respond → Analyze → Eradicate	Forensics and Evidence
Proactive	Respond → Log → Alert	Analyze → Eradicate if necessary	Forensics and Evidence

months while a highly skilled engineer is one who in addition to the above have managed and operated the device for at least twenty-four (24) months.

With skilled engineers managing the devices, we believe auto-response can be very accurate and effective. Because few statistics exist that illustrate the accuracy of automated response the statistics [16] generated from our analysis of one month's worth of data on NetSolve, Incorporated networks will be used for the illustration. If we include Code Red and Nimda activity, in 99.96% of the attacks, where automated response was used to mitigate the threat, the activity was malicious. Excluding large-scale worms, the attacks were malicious in 95.8% of auto-response uses. Of the 4.2% of the traffic that was not malicious, not all of it was desirable. Some of this traffic was peer-to-peer programs, on-line gaming, chat and other undesirable traffic that triggered alarms. The percentage of traffic that was denied that was business related was very small. It should be noted that many of these devices provide numerous different techniques for ensuring that very little, if any, legitimate traffic is denied through the use of automated response.

To determine how effective the device is in recognizing attacks we will use the most recent results [19]. In this test the worst NIDS detected 67 of 109 attacks or 61.5%, while the best detected 94 of 109 attacks for an 86.2% detection rate. Even the worst case, the 61.5% detection rate was out of the box [20] and it was reported that it would not be difficult to improve this with some custom signatures and tuning. What does all this mean? It means that the worst IDS tested can still detect at least

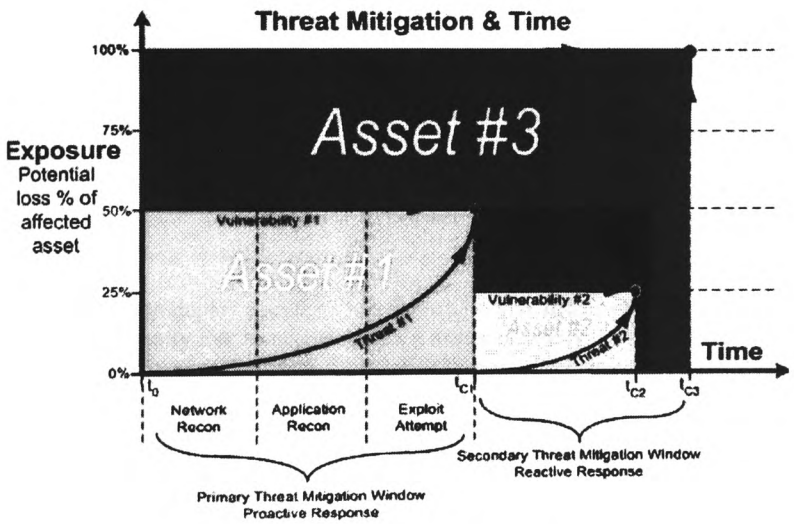


Fig. 1. Threat mitigation window [16].

61.5% of attacks. Realistically that number should be closer to 70% when a skilled engineer or technician manages the device.

The cost benefit is that the auto-response feature, when properly used, can be a very effective method of reducing the Annual Rate of Occurrence (ARO).

6.4. Adopting a cost effective staffing and support structure

The results of the studies in Section 5 demonstrate that the use of auto-response scheme produces by far better return on investment in both NIDS and HIDS deployments. A conservative estimate of 50% reduction in ARO is facilitated by the utilization of auto-response. Also, the 25% reduction in both exposure factors (EF and EFS) should also be considered a conservative estimate in the IDS deployment with auto-response and prompt incident response scheme.

The benefits of a better IDS management are reflected in the reductions in the values of the variables (see highlighted cells under ARO, EF and EFS in Table 8). The overall effect is visible in the increase in the ROI values for the ABC, Inc. IDS deployment for both the single in-house support and MSSP support schemes.

In the final analysis, attainment of a better ROI depends on a good management practice especially the use of highly skilled engineers or technicians that have a sound understanding of the technology including the inherent strengths and weaknesses to manage the IDS technology. It is also a reasonable assumption that a single in-house engineer or technician would better support IDS deployment of one NIDS and two HIDSs. On the other hand, it will be ineffective to assume that one person can support this highly dynamic technology on a continual 24/7/365 basis with active auto-response and real-time incident response for every security event. Multi-shift internal support as well as Managed Security Service Provider (MSSP) support is the preferred ways of providing definitive 24/7/365 support and real-time incident response.

7. Conclusions

The decision to deploy a security mechanism such as IDS is often motivated by the needs of security risk management. As a matter of reality, a very important but often neglected facet of intrusion detection is its *cost-effectiveness*, or *cost-benefit* trade-off. The objective of IDS is therefore to provide protection to the information assets that are at risk and have value to an organization. For IDS to be cost-effective, it should cost no more than the expected level of loss from intrusions. This requires that the IDS consider the trade-off among cost factors, which at the minimum should include development cost, the cost of damage caused by an intrusion, the cost of manual or automatic response to an intrusion, and the operational cost, which measures constraints on time and computing resources. For example, an intrusion that has

a higher response cost than damage cost should usually not be acted upon beyond simple logging.

The effectiveness of intrusion detection systems (IDS) is dependent upon an organization's deployment strategy and how well the successful implementation and management of the technology helps the organization achieve the tactical and strategic objectives it has established. One such strategic objective could be a positive Return on Investment (ROI). For organizations interested in quantifying the IDS's value prior to deploying it, their investment decision will hinge on their ability to demonstrate a positive ROI. ROI has traditionally been difficult to quantify for network security devices, in part because it is difficult to calculate risk accurately due to the subjectivity involved with its quantification. Also, business-relevant statistics regarding security incidents are not always available for consideration in analyzing risk.

In considering an implementation of IDS technology, a return on investment can be understood by analyzing the difference between annual loss expectancy (ALE) without IDS deployment and the ALE with IDS deployment, adjusted for technology and management costs. The ultimate initial goal, then, should be to prove that the value proposition (re: a benefit in the form of a quantifiable reduction in ALE) in implementing and effectively managing the IDS technology is greater than the implementation and management costs associated with deploying the IDS technology.

Finally, this paper has demonstrated that effective management methods will maximize the performance of the IDS and that a positive IDS ROI is attainable with an effective deployment technique and optimal management approach.

Acknowledgements

The contributions of Kevin Timm and David Kinn of Security Engineers at Netsolve, Inc., Austin, USA are gratefully acknowledged.

We also gratefully acknowledge Dr. Deborah Frincke's constructive suggestions on the improvement of this paper.

References

[1] K. Richards, Network based intrusion detection: a review of technologies, *Computers & Security* **18** (1999), 671–682.

[2] C. Iheagwara et al., Evaluation of the performance of IDS systems in a switched and distributed environment, *Computer Networks* **39** (2002), 93–112.

[3] C. Iheagwara et al., A comparative experimental evaluation study of intrusion detection system performance in a gigabit environment, *Journal of Computer Security* **11**(1) (2003).

[4] W. Lee et al., *Toward Cost-Sensitive Modeling for Intrusion Detection and Response*, North Carolina State University, 1999.

[5] S. Stolfo et al., Cost-Based Modeling for Fraud and Intrusion Detection Results from the JAM Project, Technical Report, Columbia University.

[6] S.A. Butler, Security attribute evaluation method: A cost-benefit approach, in: *Proceedings of the International Conference on Software Engineering*, Orlando, FL, 2002.

[7] H. Wei et al., Cost benefit analysis for network intrusion detection systems, in: *Proceedings of the CSI 28th Annual Computer Security Conference*, Washington, DC, October 2001.

[8] C. Irvine et al., Toward a taxonomy and costing method for security metrics, in: *Proceedings of the Annual Computer Security Applications Conference*, Phoenix, AZ, 1999.

[9] Cohen et al., A preliminary classification scheme for information system threats, attacks, and defenses; a cause and effect model; and some analysis based on that model, *Sandia National Laboratories*, September, 1998.

[10] Federal Information Processing Standards, *Guideline for the Analysis of Local Area Network Security*. National Institute of Standards and Technology, FIPS PUB 191, November 1994; http://www.itl.nist.gov/fipspubs/fip191.htm.

[11] R. Clarke, Computer matching by government agencies: The failure of cost/benefit analysis as a control mechanism, *Information Infrastructure and Policy 4*, 1 (March) (1995); http://www.anu.edu.au/people/Roger.Clarke/DV/MatchCBA.html.

[12] NIH's Cost-Benefit Analysis Guide for NIH IT Projects. Available at: http://wwwoirm.nih.gov/itmra/cbaguide.doc.

[13] A. Beattie et al., Timing the application of security patches for optimal uptime, in: *Proceedings of LISA'02: Sixteenth Systems Administration Conference*, USENIX Association, 2002.

[14] R.D. Silverman, A cost-based security analysis of symmetric and asymmetric key lengths, *RSA Laboratories Bulletin 13* (April) (2000).

[15] L. Gordon et al., Return on information security investments: Myths vs. realities, *Strategic Finance Magazine* (Nov.) (2002); http://www.strategicfinancemag.com/2002/11i.htm.

[16] T. Kevin et al., CTM, Technical Report, Netsolve, Inc., Austin, USA, 2002.

[17] C. Iheagwara, The effect of intrusion detection management methods on the return on investment, *Computers & Security Journal* (October) (2003) (accepted).

[18] H. Debar et al., Towards a taxonomy of intrusion-detection systems, *Computer Networks* **31** (1999).

[19] http://www.silicondefense.com/software/acbm/speed_of_snort_03_16_2001.pdf.

[20] http://www.nss.co.uk/Articles/IntrusionDetection.htm.

Appendix 10

"Intrusion Detection Systems in Large Organizations: Strategies for Effective Deployment and Sustenance." In: Proceeding of the International Conference on Industrial Engineering and Engineering Management (IE&EM'2003), Shanghai, China, on December 6-8 2003.

Proceedings of IE&EM'2003
2003 International Conference of Industrial Engineering and Engineering Management:
Global Industrial Engineering in e-century
Dec. 6-8, 2003. Shanghai, P. R. China

RID: D042

NTRUSION DETECTION SYSTEMS IN LARGE ORGANIZATIONS: STRATEGIES FOR EFFECTIVE DEPLOYMENT AND SUSTENANCE

Charles Iheagwara [1] Andrew Blyth [2]

[1.] Una Telecom, Inc., 4640 Forbes Boulevard, Suite 200, Lanham, MD 20706, USA Email: iheagwarac@aol.com

[2.] School of Computing, University of Glamorgan, Pontypridd, CF 37 1DL, Wales, UK Email: ajcblyth@glam.ac.uk

BSTRACT

'is paper discusses the challenges facing IDS deployments in ge-scale infrastructures. Further, the paper discusses emerging chitectural and structural approaches to intrusion detection stems (IDS) design, as well as the performance issues sociated with implementations in complex, inter-dependent frastructures and distributed architectures. The paper suggests ategies to enhance the performance of the IDS and support fective decision-making about which techniques are propriate for the management of the IDS in a given vironment.

INTRODUCTION

A Intrusion detection system (IDS) is a security system that nitors computer systems and network traffic and analyzes that ffic for possible hostile attacks originating from outside the ganization and also for system misuse or attacks originating m inside the organization. Thus, the main task of intrusion tection systems is defense of a computer system by detecting d possibly repelling attacks it. Detecting hostile attacks pends on the number and type of appropriate actions. Intrusion vention requires a well-selected combination of "baiting and pping" aimed at both investigations of threats. Diverting the truder's attention from protected resources is another task. th the real system and a possible trap system are constantly nitored. Data generated by intrusion detection systems is efully examined (this is the main task of each IDS) for tection of possible attacks (intrusions).

ce an intrusion has been detected, the IDS issues alerts tifying administrators of this fact. The next step is undertaken her by the administrators or the IDS itself, by taking vantage of additional countermeasures (specific block functions to terminate sessions, backup systems, routing connections to a system trap, legal infrastructure etc.) – following the organization's security policy (Fig.1).

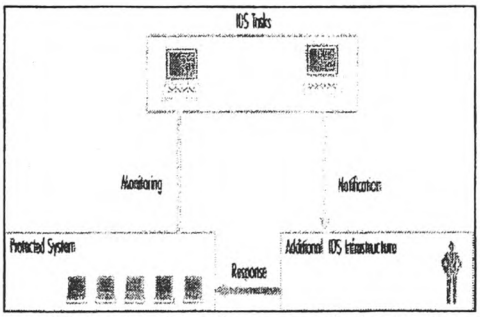


Figure 1 Intrusion detection system infrastructure [1]

Among the various IDS tasks, intruder identification is one of the fundamental ones. It can be useful in the forensic research of incidents and installing appropriate patches to enable the detection of future attack attempts targeted on specific persons or resources. Intrusion detection may sometimes produce false alarms, for example as a result of malfunctioning network interface or sending attack description or signatures via email.
Among the many challenges facing the IDS is the complex networked environment it operates in. This is compounded in large-scale deployments where various technical and operational issues bug the IDS performance. To effectively deploy the IDS in such complex and networked environments requires a broad understanding of computer security and good product delivery. As the information technology landscapes and infrastructures become more and more complex so also has the performance effectiveness of the IDS diminished in such environments [2].

S deployment in large companies presents several unique acles. The most obvious difference between small and large erprise implementations is the number of endpoint machines t must be protected. More computers, servers, and network gments mean a more complicated setup and a longer allation time. Smaller institutions, by definition, have less ices and options about where to strategically install the IDS. contrast. Larger institutions must often spend days or even eks deciding on the optimal placement of IDS agents, nagers, and IDS configuration groupings.

nother well-known issue facing big companies involves alability and the agent/console ratio. Depending on how many ployees will monitor the IDS managers for output, the ployee skill and comfort level, the number of intrusion alerts r minute, the IDS software implemented, and several other tors, the ideal agent/console ratio can vary from 5:1 to 50:1. hough others may claim that these numbers are low, our erience indicates that even the high end of this spectrum is ely achieved. Unfortunately, thorough planning in this stance may not help the situation because of the many determined and unpredictable factors that influence the timal ratio. Different IDS solutions scale differently in ifferent environments and situations. Regardless of how this ue is resolved, it must be addressed: several organizations' DS units and entire managed security services providers have iled as a result of their inability to scale.

he rest of the paper is organized as follows. In Section 2, we iscuss IDS structure and architecture. In Section 3, we present e IDS challenges in large-scale distributed infrastructures and iscuss mitigation strategies Section 4. We conclude our iscussion in Section 5.

2 THE STRUCTURE AND ARCHITECTURE OF NTRUSION

1detection systems

t the component level, an intrusion detection system always as its core element - a sensor (an analysis engine) that is sponsible for detecting intrusions. This sensor contains ecision-making mechanisms regarding intrusions. Sensors eceive raw data from three major information sources: own IDS nowledge base, syslog and audit trails. The syslog may include, r example, configuration of file system, user authorizations etc. his information creates the basis for a further decision-making rocess.

he sensor is integrated with the component responsible for data ollection (Fig.2) — an event generator. The collection manner s determined by the event generator policy that defines the iltering mode of event notification information. The event enerator (operating system, network, application) produces a olicy-consistent set of events that may be a log (or audit) of ystem events, or network packets. This, set along with the olicy information can be stored either in the protected system r outside. In certain cases, no data storage is employed for xample, when event data streams are transferred directly to the nalyzer. This concerns the network packets in particular.

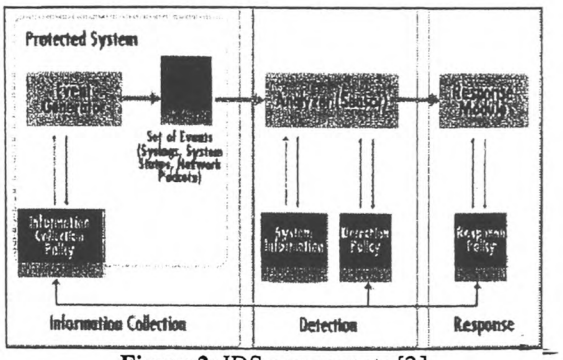


Figure 2: IDS components [3]

The role of the sensor is to filter information and discard any irrelevant data obtained from the event set associated with the protected system, thereby detecting suspicious activities. The analyzer uses the detection policy database for this purpose. The latter comprises the following elements: attack signatures, normal behavior profiles, and necessary parameters for example, thresholds. In addition, the database holds IDS configuration parameters, including modes of communication with the response module. The sensor also has its own database containing the dynamic history of potential complex intrusions (composed from multiple actions).

Intrusion detection systems can be arranged as either centralized (for example, physically integrated within a firewall) or distributed. A distributed IDS consists of multiple Intrusion Detection Systems (IDS) over a large network, all of which communicate with each other. More sophisticated systems follow an agent structure principle where small autonomous modules are organized on a per-host basis across the protected network [4]. The role of the agent is to monitor and filter all activities within the protected area and — depending on the approach adopted — make an initial analysis and even undertake a response action. The cooperative agent network that reports to the central analysis server is one of the most important components of intrusion detection systems. DIDS can employ more sophisticated analysis tools, particularly connected with the detection of distributed attacks [5]. Another separate role of the agent is associated with its mobility and roaming across multiple physical locations. In addition, agents can be specifically devoted to detect certain known attack signatures. This is a decisive factor when introducing protection means associated with new types of attacks [6]. IDS agent-based solutions also use less sophisticated mechanisms for response policy updating [7].

One multi-agent architecture solution, which originated in 1994, is AAFID (Autonomous Agents for Intrusion Detection). It uses agents that monitor a certain aspect of the behavior of the system they reside on at the time. For example, an agent can see an abnormal number of telnet sessions within the system it monitors. An agent has the capacity to issue an alert when detecting a suspicious event. Agents can be cloned and shifted onto other systems (autonomy feature). Apart from agents, the system may have transceivers to monitor all operations effected by agents of a specific host. Transceivers always send the results of their operations to a unique single monitor. Monitors receive information from a specific network area (not only from a single

ISs), which means that they can correlate distributed information. Additionally, some filters may be introduced for data selection and aggregation [8].

3 THE CHALLENGES FOR LARGE-SCALE DISTRIBUTED INFRASTRUCTURES

Intrusion detection for emerging large-scale distributed systems (e.g. global companies and virtual enterprise networks) faces a variety of difficult challenges. The most important ones can be summarized as:

3.1 Multiple attack scenarios

The anatomy of an intrusion is composed of increasingly complex attack scenarios. An attack scenario consists in a logical sequence of actions that are applied for reaching a particular strategic goal (e.g. getting confidential information). These actions are typically applied on different hosts in a network and by using a variety of tools. Moreover, a variety of different attack scenarios are possible to reach the same goal. There is a need for dynamically linking local individual events to global attack strategies in order to provide pro-active and adaptive intrusion monitoring

3.2 Performance in complex infrastructures

Large distributed networks of systems need scalable IDS approaches for which performance is becoming an important attribute. This includes issues of timeliness of local event monitoring and communication of contextual data between nodes as well as of trust relationships between the nodes.

3.3 Communication protocols

Most of the individual techniques are more suitable for local event monitoring and analysis. Globally co-ordinate attack strategies require integration of methods and aggregation of disparate information sources. The critical issue lies in defining the high-level communication protocol to allow different methods of IDS to contribute to the intrusion detection process.

3.4 Integration with network management system

IDS system methods must be better integrated with exiting network management systems if their widespread adoption in industry is to be guaranteed. One reason is that this should facilitate their maintenance/upgrades and a more coherent audit/log data management. IDS are one mechanism to respond to new business dependability/survivability needs. It is as yet unclear how to integrate IDS with other dependability mechanisms (e.g. fault tolerance, recovery mechanisms) in a wider information risk and security policy context.

3.5 Implementation issues

Acknowledging the need for IDS protection, and subsequently choosing the IDS that best fits the company's needs are important steps in the quest for overall information security. However, these steps only complete the initial stages of a thorough IDS implementation process. After selecting and purchasing the optimal IDS, a company must properly and efficiently deploy it throughout the organization.

The first step in a well-planned and thorough deployment should be to design an IDS strategy and then express it in the context of an IDS policy. This policy document serves as a guide for the implementation process, answering questions such as:

1. Will network traffic restrictions be tight or loose?
2. Who will be authorized to make changes to the IDS policy or configurations?
3. On which machines will an IDS installation be required?
4. How frequently will IDS logs undergo analysis?

The planning and coordination required in creating this policy will reinforce the communication between company management and security officials. At the same time, this will allow both organizational units to identify and resolve conflicts before they become obstacles to successful IDS deployment. Organizations should incorporate this policy into their overall security policy or company rules and regulations.

3.5.1 Installing: After the IDS policy is set, we move to the next logical step, installation. Installing an IDS system typically begins with installation of the IDS manager. Generally, the procedures for this installation are similar to those for most software: insert the CD and locate the executable. Although this process is straightforward, Murphy's Law suggests that it infrequently runs smoothly. The installation wizard might freeze, the installation options available might sound complicated and unfamiliar, or a particular .DLL file might not unpack correctly. Though these problems are comparatively miniscule in the overall process of IDS deployment, they must still be resolved prior to moving on to the next phase of deployment.

Upon completion of manager installation, the IDS tool must be distributed to agents through one of a number of different methods. Most installation problems occur in this portion of IDS deployment. The major obstacles that arise in distributing IDS agents relate to communication between the agents and managers. These problems often surface in the following areas:

1. The trust relationship between the systems on which agents and managers reside;
2. Communication issues with Network Address Translation; and,
3. Discrepancies in the installation process itself – i.e., the steps that must be followed to maintain establishment of a secure channel between an agent and the manager.

Companies have several available options to deal with these issues: using the system defaults, reading the manual, navigating through on-line help, calling the IDS vendor's software help desk, and/or using outside consultants. Any of these methods should ultimately result in a successful installation.

3.5.2 Configuration: Organizations must deal with the issue of setting the IDS to capture relevant data only. Every organization has different expectations and different requirements, so the default IDS settings usually need to be altered. Finding the perfect balance between a massive amount of data generation, which leads to an over-saturation of information, and a small amount of data generation, which may cause ineffective monitoring, can complicate a deployment. In general, a sophisticated IDS solution will require a sophisticated IDS configuration, so companies should budget plenty of time for thorough configuration development, tuning, and testing.

Monitoring of key segments

large scale environments there may be a requirement to place veral network intrusion detection devices in several locations the network. Network based Intrusion Detection Systems are y able to monitor network traffic on the network segment y reside on.

STRATEGIES FOR THE EFFECTIVE SUSTENANCE F IDS PERFORMANCE

e complexity, as well as the importance of enterprise system tributed computer networks and their information resources lls for new approach in the management of intrusion nitoring.. Large theoretical and practical efforts are ncentrated on this problem today. In the Sections below, we esent demonstrated strategies that can help improve the IDS rformance in large-scale infrastructures.

Defining attacks

o effectively defend the network, one must delineate what a rmal or abnormal traffic is. Since intrusion detection systems al with hacking breaches, one must clearly define and aracterize all the possible attacks that are presently known or nceivable. Towards this, intrusion attacks have been presented the scheme of Kumar [9] and can be represented by an event series of events. It is the relationship of these events to one ther that provides the basis for recognizing differing attack pes. Under the attack taxonomy, intrusions fall into two tegories: namely misuse and anomalous behavior. Misuse mprises attacks that are already known and whose behavior n be specified while anomalous behavior describes attacks volving unusual use of the system resources. In our periments the attack set described below falls under the active suse attack type.

Recognizing Attacks: Symptomatic Precursors

amining systems for any abnormal behavior could lead to cognition of possible attacks. This may be helpful in detecting acks. In most cases, any attempt to take advantage of faults in ganization security systems may be considered as an attack d this is the most common symptom of an intrusion. However e organization itself may "facilitate" the task of attackers, ng tools, which aid in the process of securing its network – so alled security and file integrity scanners. They operate either cally (assisting system administrators during scanning) or motely but may also be deliberately used by intruders. ertain scanning tools often double-edged swords available for oth the users and hackers are good for monitoring system and twork activities e.g. file integrity scanners and known lnerability scanners. It is worthy to note the following:

- Detection of file integrity scanners. The available file integrity testing tools operate in a systematic manner so that it is possible to use modeling techniques and specialized tools for detection purposes, for example the anti-SATAN software, Courtney.

A good correlation between scanning and usage is required – scanning for flaws may further use a service featuring such flaws, this may be a precursor of an attack to come.

4.3 Suspicious network activity

An intruder actually trying to compromise a system often uses a large number of exploits and makes many unsuccessful attempts. His activities differ from those of the user working with the system [10]. Any penetration-testing tool should be able to identify suspicious activities after a certain threshold has been exceeded. Then, an alert may be produced and diffused. This passive technique allows detection of intruders without discovering a clear picture of the event (exploits involved, tools, services, software configuration, etc.), by only quantitatively examining network activities.
Passive methods used in intrusion detection are driven from databases of recurrent attack signatures that should consider the following technical aspects:

- Repetition thresholds to help distinguish between legal and suspicious activities (that trigger alerts). Network activities can be identified using multiple parameter values derived, for example, from the user profile or Session State.
- Time between repeat instances – a parameter to determine the time to elapse between consecutive events, for example, an activity is to be considered suspicious if within a two-minute interval, three consecutive unsuccessful login attempts are made.
- Constructing a database of repetitive attack signatures. An attack may involve neutral activities (mostly in the reconnaissance phase) and/or those misleading the IDS defense devices. In such a case, construction of an attack signature may be impossible or very difficult.

4.4 Incompatible or illegal commands

Network services/protocols are documented in a precise manner and use determining software tools. Any incompatibility with known patterns (including typical human errors such as misprints occurring in network packets) may be valuable information to detect services that are possibly being targeted by an intruder.
If the system audit facility uses, for example, send mail relaying, then the relevant log sequence behaves in a regular and predictable manner. However, if the log indicates that a specific process has given illegal commands, it might be a symptom of either a non-malicious event or a spoofing attempt.
The examining of hostile attempts may include:

- Detection of attempts to recover mistyped commands or answers followed by re-launching them,
- Detecting several failed attempts to observe syntax protocols followed by successful ones,
- Detecting adaptive learning attempts to capture errors committed by the same object (service, host). After a certain period, these errors will cease.

Abnormal attributes

The most frequent cases are the ones where one is expected to deal with a set of attributes of packets or specific requests for services. It is possible to define the expected attribute pattern. If encountered attributes do not match this pattern, this may indicate a successful intrusion or intrusive attempt.

- Calendar and time attributes
- System resource attributes.
- Packets with unexpected TCP acknowledgement settings.
- Service mix attributes.

Odd System Behavior

A potential intruder may design its malicious activity with side effects that will cause odd behavior of the system. Monitoring such side effects is difficult since their location is hardly detectable. Below there are some examples of:

- Unexplained problems with system hardware or software, for example server down, particularly daemons not running, unexplained system restart attempts, changes to system clock settings.
- Unexplained system resource problems: file system overflow; abnormal consumption of CPU usage.
- Odd messages from system daemons, system daemons not running or disturbed (particularly superuser daemons and those designed to monitor the system state, for example Syslog). Such symptoms are always suspicious.
- Unexplained system performance problems (routers or system services, for example long server response times).
- Unexplained user process behavior, for example unexpected access to system resources.
- Unexplained audit log behavior. Audit logs that shrink in size (unless intentionally reduced by the system administrator).

Effective Configuration Strategies

To maximize the IDS effectiveness, there is the need to develop lifecycle operational Strategies that enhances the performance indexes.

Reviewing the deployment policy: The IDS administrator needs to review the current IDS deployment policy detailing the organization's approach to intrusion detection in general. The policy determines which strategy to employ, hence determine what can be done to help improve IDS performance. This could for instance stipulate the methods to monitor attacks. Possible options include:

1. To monitor for all attacks, regardless of what systems are prevalent in an organization; for example, looking for RPC exploits in a Microsoft environment.
2. To monitor only for attacks that would be relevant to the network environment, such as configuring the NIDS to detect all Microsoft exploits in an all Microsoft environment;
3. To monitor all vulnerabilities for a particular service regardless environment, such as detecting all HTTP exploits in an IIS-only environment.

4.7.2 Filtering Signatures: After determining the strategy, IDS engineer should trim the amount of attacks that the IDS will look for. The NIDS default configuration may monitor for potential attacks that are not are not relevant to the environment that. This can lead to wasted resources and, thus, to inefficiencies. For example, if the company is a pure Windows environment there is no real reason to look for RPC exploits. Signature trimming can remove many unnecessary signatures at a time. For example, if the system is running Snort, the admin can simply edit the snort.conf file and remove the entire rules file (i.e. rpc.rules, x11.rules). To get more granular, the administrator should look at the more common services (i.e. HTTP, FTP, SMTP) and see if the attack signatures they are looking for match the services that the company runs. In this context, it makes sense to look for signatures that match software vendors that are on the network. For example, if the company uses FTP servers, but none of them are running wu-ftp, the NIDS does not need to be configured to look for wu-ftp exploits.

4.7.3 Traffic Filtering: Most NIDSs have some sort of filtering function that allows certain types of traffic to be disregarded. There are a couple instances when this type of filtering may be of value. Firstly, if there is a server or subnet that generates a lot of traffic that does not need to be monitored. One form of this type of traffic would be multicast traffic, which is usually some type of streaming media. Some switch vendors may be able to filter the traffic before it gets to the NIDS. This would be the preferred approach, as the NIDS will not have to unnecessarily process data that has already been determined to be harmless. Secondly, this type of filtering may be useful if there are servers or subnets that generate traffic that is encrypted or that the NIDS otherwise can't decode. Since this type of traffic would need to be to thrown away, there is no reason to search the packets in the first place. Also, most NIDS are installed with two NICs (Network Interface Cards), one for monitoring a particular network segment and one for management. Depending on how the network is designed there could be a lot of erroneous packets hitting the NIDS management NIC, even in a purely switched environment.

4.7.4 Load Balancing: Another type of filtering could be done with some type of hardware device. This would allow traffic to be split up and sent to a NIDS farm (a logical grouping of multiple physical NIDS to handle high bandwidth networks). This is the same idea as DNS or Web server load balancing that is currently used on many networks. There are products called IDS Balancers, which are aimed at NIDS load balancing. With these types of devices, 100Mbs+ of bandwidth can be distributed over multiple sensors in the NIDS farm. In a basic distribution scenario, the traffic would be evenly distributed among each NIDS in the NIDS farm. This means that each NIDS should get an equal amount of traffic. Since today's NIDSs are more stateful than previous generations, each NIDS would have to get traffic distributed to it based on sessions (conversations), not individual packets. This way the NIDS can watch the entire conversation and be aware of any attacks or anomalies. Each NIDS would have the same policies, so that they would catch the same exploits and anomalies. With devices such as NIDS load

ancers and layer-7 switches, the traffic could be filtered fore it is sent to the NIDS farm.

7.5 **Changing Default Timeouts**: Depending on the NIDS ng deployed, there maybe options to configure how may nections are tracked. Altering this configuration will affect e amount of memory used to store this information, as well as e amount of CPU cycles used to search this information. When se timeouts are defined, a certain amount of memory is set de for these tables, regardless of whether or not they are filled. y doing some traffic analysis on the network, the settings could e changed to make the NIDS more efficient when watching a rticular network segment.

ere may be options available to define timeouts for particular otocols. These may be as generic as TCP and UDP timeouts, hey may be as granular as HTTP, DNS and SMTP timeouts. pending on the type and amount of a particular protocol on e network, the NIDS administrator may need to specify a fferent timeout than the default.

7.6 **Reducing the Traffic Volume:** This options fall into the tegory of reducing the volume of information that the twork-based IDS is required to filter without reducing or ming off the information that employees or customers need.

7.7 **Use information caches**: Both proxy and client caches are eful in the effort to reduce traffic. "Web Caching is the act of ring copies of Web pages on a 'local' system. If the same ges are requested at a later time, and the cached copy is still id, there is no need to contact the origin server again."[7] The che could be on a separate server that is pointed to by client b browsers and used as a proxy, or it can be the built in cache at is part of most web browsers. By implementing caches, ers will download information from a web page, the formation will be checked by the network-based IDS, and it ll be displayed to the user. In addition, the downloaded formation will be stored in the cache. The next time that formation is requested, it comes from the cache and does not ve to be checked by the network-based IDS. This only works the cache server is positioned after the network-based IDS in e traffic flow, or if users are using the cache capability in their b browsers. Web browsers can be configured with how much sk space to use to store downloaded web pages. If this value is o low, it may need to be raised to be of any benefit.

7.8 **Remove unused protocols:** Preconfigured, out-of-the-box mputer systems normally have multiple network protocols stalled that are not needed. Uninstalling those protocols will duce the load on a network and on network-based IDS. If a otocol is not running on a network then obviously the twork-based IDS will not have to spend cycles analyzing its ckets. Removing protocols that the network-based IDS will or analyze will not provide as much benefit as removing those otocols that it will analyze. It is still a good security practice to move them though. Most network-based IDS systems only alyze IP traffic and not AppleTalk, NetBEUI, or IPX. twork-based IDS tools normally consist of a packet collection gine and a packet analysis engine. The packet collection gine will pull all packets off of the network, including pleTalk and IPX packets, but it will not send the packets taining protocols that the analysis engine cannot analyze to analysis engine itself. So pulling the unused protocols off the network means the packet analysis portion of the network-based IDS tool will have fewer packets to process.

5. CONCLUSIONS

This paper has discussed the challenges associated with deployment of the IDS in large-scale infrastructures and the factors that impede their performance. The effectiveness of the IDS is dependent upon an organization's deployment strategy and how well the management of the IDS technology helps the organization achieve the tactical and strategic objectives it has established. The strategies to attain such objectives have been outlined and discussed in Section 4 of this paper. Finally, it has been shown how the use of effective configuration techniques can reduce the problems associated with the IDS performance in large deployments.

REFERENCES

[1] P. Dorosz, P. Kazienko, Systemy wykrywania intruzów, VI Krajowa Konferencja Zastosowan Kryptografii ENIGMA 2002, Warsaw 14-17 05.2002. , p. TIV 47-78,

[2] Iheagwara C. and Blyth A, "Evaluation of the performance of IDS systems in a switched and distributed environment," *Computer Networks*, 39 (2002) 93-112

[3] E. Lundin, E. Jonsson, Survey of research in the intrusion detection area, Technical report 02-04, Department of Computer Engineering, Chalmers University of Technology, Göteborg January 2002,

[4] C. Krügel, T. Toth, Applying Mobile Agent Technology to Intrusion Detection, ICSE Workshop on Software Engineering and Mobility, Toronto May 2001.

[5] C. Krügel, T. Toth, Distributed Pattern Detection for Intrusion Detection, Conference Proceedings of the Network and Distributed System Security Symposium NDSS '02, 2002,

[6] J.S. Balasubramaniyan, J.O. Garcia-Fernandez, D. Isaco, E. Spafford, D. Zamboni, An Architecture for Intrusion Detection using Autonomous Agents, 14th IEEE Computer Security Applications Conference ACSAC '98, December 1998, pages 13-24.

[7] D.J. Ragsdale, C.A. Carver, J.W. Humphries, U.W. Pooh, Adaptation techniques for intrusion detection and intrusion response systems, Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 2000, pages 2344-2349.

[8] E.H. Spafford, D. Zamboni, Intrusion detection using autonomous agents, Computer Networks 34, 2000, pages 547-570.

[9] S. Kumar. Classification and Detection of Computer Intrusions. Department of Computer Sciences, Purdue University. Ph.D. Dissertation, 1995.

[10] G. Mansfield, K. Ohta, Y. Takei, N. Kato, Y. Nemoto, Towards trapping wily intruders in the large, Computer Networks 34, 2000, pages 659-670.

Appendix 11

"Intrusion Detection Challenges: charting the course for research and development" In: Proceeding of the International Conference on Industrial Engineering and Engineering Management (IE&EM'2003), Shanghai, China, on December 6-8 2003.

Proceedings of IE&EM'2003
2003 International Conference of Industrial Engineering and Engineering Management:
Global Industrial Engineering in e-century
Dec. 6-8, 2003. Shanghai, P. R. China

RID: D032

INTRUSION DETECTION CHALLENGES: CHARTING THE COURSE FOR RESEARCH AND DEVELOPMENT

Charles Iheagwara [1] Andrew Blyth [2]

[1] Una Telecom, Inc., 4640 Forbes Boulevard, Suite 200, Lanham, MD 20706, USA Email: iheagwarac@aol.com

[2] School of Computing, University of Glamorgan, Pontypridd, CF 37 1DL, Wales, UK Email: ajcblyth@glam.ac.uk

ABSTRACT

The intrusion detection system (IDS) technology is very powerful. It provides an incredible view into the security problems facing an organization. However, it still requires considerable refinements to eliminate the weaknesses in currently available products. Some of the weaknesses that are considered short-term i.e. scalability, hierarchical reporting, and dynamic remote updates are already being addressed by vendors and some of the solutions are in the alpha- or beta-testing stage. The long-term weaknesses, which by and large are metrics for IDS performance, are being addressed as research topics in and of themselves.

In this paper, we explore the new areas of IDS development and chart the course for future research. We propose standard frameworks to conceptualize the most appropriate remedies to current design problems given the present or projected availability of appropriate countermeasures. We explore how the proposed frameworks can be translated into design standards for IDS products in the most consistent and meaningful way.

Keywords: Intrusion detection systems, Intrusion challenges, IDS Design.

I. INTRODUCTION

An intrusion detection system (IDS) is a security system that monitors computer systems and network traffic and analyzes that traffic for possible hostile attacks originating from outside the organization and also for system misuse or attacks originating from inside the organization.

IDS, much like the security industry itself, have grown rapidly over the past few years. These tools have become essential security components - as valuable to many organizations as a firewall. However, as in any environment, things change. As networks and crackers evolve and grow rapidly, demanding that security tools keep up, the IDS faces several daunting but exciting challenges in the future and are sure to remain one of the best weapons in the arena of network security.

Among the many challenges is the development of commercial IDS products to suite the needs of today's complex networked environments. To effectively deploy the IDS in such complex and networked environments requires a broad understanding of computer security and good product delivery. As the information technology landscapes and infrastructures become more and more complex so also has the performance effectiveness of the IDS diminished in such environments [1]. The complexities have often resulted into the most significant obstacle to the success of an information security improvement initiative due to lack of management support. In surveys conducted by security trade magazines [2], lack of management support was cited as one of the principal barriers to effective information security.

Within the IDS market place are two broad categories of products: Host-based and Network-based. Commercially available IDS products (Table 1) are classified according to their approach to intrusion detection with all being either host or network-based. None of the products integrate host-based and network-based intrusion detection capabilities and a few integrate security assessment capabilities with basic IDS

functionality, such as audit trail analysis and malicious software protection.

The insights and operational experience gained from the implementation of the currently available IDS products demonstrates the need for product enhancements. In this regard, results of previous research studies on performance effectiveness could be helpful in evolving an effective design approach and charting the course for new products development. Crucial to this is the results of those research studies that established a link between IDS performance in technical, operational and cost/economic terms and product architectures.

With so many studies on IDS technologies, those that are performance-based have leaped forward to take the center stage as developers race to refine existing products. IDS performance studies [1,3,4] measure the effectiveness of the IDS using a variety of metrics for different environments.

Table 1. IDS products

Vendor	Product Name	Approach to ID	Platform
ISS	RealSecure	Network-based Packet capture, signature analysis, and real-time playback.	UNIX And NT
Cisco	NetRanger	Network-based. Passive network monitor with packet filtering router.	UNIX
Security Dynamics	Kane Security Monitor	Host-based. Passive ID capabilities with assessment functions	NT
DMW Worldwide	HostCHECK	Host-based. Passive ID capabilities (audit trail analysis and file checksums) and assessment functions.	UNIX
MEMCO	SessionWall	Network-based. Real-time connection and playback.	NT

Richards [3] evaluates the functional and performance capabilities of the industries' leading commercial type IDS. In the areas tested, the performance of the IDS was rated based on their distinctive features, which were characterized into different performance indexes. The research work represented a new direction for ID systems in that it moved the focus away from scientific concepts research to performance evaluation of the industries' best products. However, the study was limited to a small proto design isolated and non-switched network which did not reveal the impact of packet switching on the accuracy and ability to capture attack packets in their entirety. Iheagwara et al. [1] expands this effort with an evaluation study of the effect of deployment techniques on IDS performance in switched and distributed system. The study demonstrates that monitoring techniques could play an important role in determining the effectiveness of the IDS in a switched and distributed network.

In an experimental evaluation study of intrusion detection system performance in a gigabit environment, Iheagwara et. al. [4] examines the system benefits of using a single Gigabit IDS sensor instead of multiple Megabit sensors for a wide range of defined system attacks, network traffic characteristics, and for their contexts of operational concepts and deployment techniques. The study established the relationship between traffic parametric values and the IDS performance for specified environments.

The results of cost-benefit model and analysis studies of IDS deployments, although relatively few shades some light on the integral economic building blocks that could serve as a guide for acquisition and deployment of the products. Lee et. al. [5] studies the problem of building cost-sensitive intrusion detection models. For intrusion detection, Irvine [6] defines auditing of network control functions in intermediate nodes, and rule-based network intrusion systems in the total subnet as the mechanisms. Irvine also discusses the costs of those security services and mechanisms. Yet to be developed, is a cost model to serve as a guide for new product design.

While these studies characterize areas of design and implementation weaknesses, none have provided the IDS developers with the data and techniques necessary to create truly "next-generation" intrusion detection algorithms and tools. For instance, none of the performance evaluation studies [1,3,4,] that explore the relationship between deployment techniques and attack system variables and the performance of the IDS; and the studies [5, 6] on cost models that investigate the cost-benefit/sensitive analysis for intrusion detection deployment presents developers with any concrete guide on how to translate research results into practical design tools.

Going by the results of the studies, the need to establish a uniform framework for new products development or the refinement of existing ones using the results of research works and experiences gained from IDS implementations cannot be overstated.

Therefore, this paper will seek to chart a course for the future development of new IDS products by drawing from the field experiences of the authors in the operation of the IDS technology. Further, we will demonstrate that the concepts proposed here could reasonably be associated with well thought out propositions on the requirements and functions that must be satisfied in order for new products to be implemented successfully in all design lifecycle.

The rest of the paper is organized as follows. In Section 2, we present a development framework for future product design. In Section 3, we present the basic elements that will make for reliable and effective IDS and then propose research and development directions to realize this in Section 4 and conclude our discussion in Section 5.

2. FRAMEWORK FOR IDS DEVELOPMENT

As with other security and monitoring products, intrusion detection systems functions as one element of a corporate security policy. Successful intrusion detection requires that a well-defined policy on IDS development be set up to ensure that intrusions are handled according to corporate security policy guidelines. The development of new IDS products requires innovative approach on how the IDS is designed, deployed and maintained. Without this, the successful use of this technology will be short-lived.

In order to develop any innovative approach towards IDS product development within various overriding security requirements, considerations will begin to focus on defects, remedial design measures and cost trade-offs, including the cost of long-term maintenance and reliability requirements. The approach will take into account the fact that often conceptualized product development model is different from the real world development model. Theoretically, the assumptions made under conceptualized product development model might be correct but because of difficulties in vision and execution differences could exist with the real world development model. Consequently, the development of standard design frameworks should draw from the lessons learned from current implementation failures; should be valuable in directing remedial and redeveloping efforts and should aim to eliminate known defects in current commercial products.

The conceptualized product development model always takes into account the desire of both developers and users that the design of the IDS incorporates the basic elements that will maximize the **technical effectiveness of the IDS**. This often translates into IDS vendors attempting to use brute force methods to correctly detect a larger spectrum of intrusions than their competitors. However, the goal of catching all attacks has proven to be a major technical challenge. After more than two decades of research and development efforts, the leading IDSs still have marginal detection rates and high false alarm rates, especially in the face of **stealthy** or **novel** intrusions. This goal (of catching all attacks) is also impractical for IDS deployment, as the constraints on time (i.e., processing speed) and resources (human/computer) may become very restrictive.

The Frameworks that will set the stage for product development should be formulated in broader terms and should be drawn from the difficulties in the current implementation. Despite the fact that the technologies of IDS commercial products are laden with multiple problems many vendors are still developing new products at a furious pace. Some of the weaknesses that are considered short-term i.e. scalability, hierarchical reporting, and dynamic remote updates are already being addressed by vendors and some of the solutions are in the alpha- or beta-testing stage. Compounding the situation are the problems posed by the huge number of applications that have unknown and undocumented holes. For the typical IDS the main way of detecting such attacks is to use attack signatures to learn what a weakness looks like and detecting attempts to trigger the hole from outsiders. This is fairly limited, technically.

The other pitfalls [7] include:

1. The issues of variant signatures,
2. False positives and negatives alerts,
3. Data overload,
4. Interoperability,
5. Difficulties to function effectively in switched environments, and
6. Scalability issues.

A description of the pitfalls is given elsewhere [4].

Thus, the frameworks should provide indication of possible future directions, by addressing the following pitfalls-related questions:

- How can IDSs be evaluated?
- Do current systems, particularly anomaly-based IDSs, produce too many false positives?
- Can network IDSs be scaled to the ever increasing bandwidth of networks?
- Can intrusion detection be used to detect unknown attacks?
- How can the reports of multiple intrusion detection systems be combined to detect attacks that span multiple locations and subsystems, or that plays out over time?
- Can intrusion detection and automated response be integrated, as necessary to cope with rapidly spreading attacks or situations where human response is not possible?
- Will intrusion detection technology be useful for potentially devastating attacks, such as a malicious worm that in the absence of a defense would spread over the Internet?

The formulation of standard frameworks would also entail adopting a complementary approach to encapsulate both the good and bad experiences in the current implementations into empirical design rules.

The frameworks should detail how empirically derived solutions could be applied to specific design processes and should provide the necessary safeguards to guarantee that following one design principle will not lead to violating another. The following perspectives that border on IDS implementation should serves as the frameworks for future development directions and the evolution of design standards:

1. IDS technology itself is undergoing a lot of enhancements and the technology has not reached a level where it does not require human intervention. Of course today's IDS technology offers some automation like notifying the administrator in case of detection of a malicious activity, shunning the malicious connection for a configurable period of time, dynamically modifying a router's access control list in order to stop a malicious connection etc. But it is still important to monitor the IDS logs regularly to stay on top of the occurrence of events. Monitoring the logs on a daily basis is required to analyze the kind of malicious activities detected by the IDS over a period of time. Today's IDS has not yet reached the level where it can give historical analysis of the intrusions detected over a period of time. This is still a manual activity. It is therefore important to use this as a framework for technology refinement.

2. The success of an IDS implementation depends to a large extent on how it has been deployed. A lot of planning is required in the design as well as the implementation phase. In most cases, it is desirable to implement a hybrid solution of network based and host based IDS to benefit from both. In fact one technology complements the other. However, this decision can vary from one organization to another. A network based IDS is an immediate choice for many organizations because of its ability to monitor multiple systems and also the fact that it does not require a software to be loaded on a production system unlike host based IDS. Some organizations implement a hybrid solution. A suitable framework here would be for organizations to clearly define their expectations and to align them with policy definitions of technological effective performance of the IDS.

3. It is important to take care of sensor to manager ratio. There is no thumb rule as such for calculating this ratio. To a large extent it depends upon how many different kinds of traffic is being monitored by each sensor and in what environment. Lot of organizations deploy at a 10:1 ratio. Some organizations go for 20:1 and some others 15:1. Therefore an important framework would be to design the baseline policy before starting the IDS implementation in order to avoid false positives. A badly configured IDS sensor may send a lot of false positives to the console and even a 10:1 or even better sensor to console ratio can be inadequate.

4. The IDS technology is still reactive rather than proactive. The IDS technology works on attack signatures. Attack signatures are attack patterns of previous attacks. The signature database needs to be updated whenever a different kind of attack is detected and the fix for the same is available. The frequency of signature update varies from vendor to vendor. A suitable framework would be to design the IDS with a periodic pull mechanism similar to what obtains with anti virus update mechanisms.

5. While deploying a network based IDS solution, it is important to keep in mind one very important aspect of the network based IDS in switched environment. Unlike a HUB based network, where a host on one port can see traffic in and out of every o ther port in the HUB, in a switched network however, traffic in and out of one port cannot be seen by a host in another port, because they are in different collision domains.

6. A network based IDS sensor needs to see traffic in and out of a port to detect any malicious traffic. In a switched environment, port mirroring or spanning is required to achieve this. One entire VLAN can be spanned to one port on which the network based IDS sensor is installed. Although this is a solution, there may be performance issues for a busy network. If all the 10/100 Mbps ports in a VLAN are mirrored to another 10/100 Mbps port in the VLAN, the IDS sensor may drop traffic, as the combined traffic of all the ports could be more than 100 Mbps. Now, with Gigabit port speed being available, this becomes a more difficult challenge. Cisco systems has an IDS module for Catalyst 6000 series switch which can sit on the switch back plane and can monitor traffic right off the switch back plane. But this solution is yet to scale to Gigabit speed. This module supports traffic only up to 100 Mbps as of now. The portability of network based IDS in a switched environment is still a concern and should be a design framework.

The above-proposed frameworks should be used to define the development of design architectures and to prove their correctness throughout the life cycle of the IDS products.

In the next Section, we chart a course for IDS design.

3. CHARTING THE COURSE FOR FUTURE DESIGNS

Generally, expectations for IDS performance revolve around accuracy and border on the stated missions of intrusion detection systems as discovery and detection tools. The IDS technology does not directly address some other important security issues such as identification/authentication, confidentiality, etc., though some of them will be integrated with the IDS technology in the near future. Given the above and the implementation failures discussed in Section 4, there is the need to accurate the development directions in terms of not only remedial design actions but also in terms of the stated mission objectives of the IDS.

While not exclusive the following could be described as potential directions for the refinement and enhancement of existing products and development of new technologies:

1. The development of data correlation tools using analysis and correlation techniques is an immediate product development challenge. This will help to resolve data correlation issues. The challenge is to devise a tool and mechanism that will reduce human errors in the correlation of data. No matter how good the IDS analyst, abuses such as slow port scans are difficult to detect, especially on large networks. Projects such as Spice and Spade [8] are working to

make this possible. Acting as anomaly detectors, they examine strange packets and look to group them using sophisticated statistical analysis. Tools like these will aid the intrusion detection analyst to pull out " needles in the haystack" and bring them to our attention. It is worth noting that it is much easier to discover and categorize patterns when you have all the relevant data in front of you, without the noise that generally follows. Furthermore, applications similar to these will be used to fine tune filters and rules in order to reduce false positives, over time providing a kind of IDS feedback system, based on administrator input and response.

2. As a functional requirement an IDS deployment should have some operational procedure behind it to gather additional information and fine-tune the network and the process. A good IDS will automate much of this process.

3. Realistic expectations are that the product should detect, in near real-time, any kinds of attempts to exploit known weaknesses, or to probe your internal network. They should also keep track of attempts to overload necessary resources. Along with this, they should perhaps sound an alarm, trigger some predefined action, and keep a good log for analysis. Many customers think that a given security product like an IDS will protect them from 100% of the "bad things." In a practical world, there are no absolutes, instead IDS can significantly reduce the risk from network-based threats, but they're not perfect.

4. A reasonable expectation is to have an IDS detect attacks against previously unknown vulnerabilities, detect slow attacks by insiders, have built-in automatic protection for your network when an attack is detected, or to be operable in "hands-off" mode.

5. With the exception of better databases of attack signatures and methods to update those databases, the single-sensor solution has gone as far as it can go. The next big step is distributed intrusion detection-taking the results from multiple sensors, deployed throughout the enterprise, and correlating them into a single "big picture" view of the network. The only way to do this today is manually, and human in the process for some time to come. But there's going to be a lot of "software assist" showing up in the next few years. This is currently happening in the firewall market is going to hit the IDS market in the next 18-24 months. There will be several products; all more or less interchangeable at their core, with differentiation based on the assorted bells and whistles bundled with the base. One area to pay close attention to is what the vendors do to take advantage of all these sensors deployed throughout the network, all with the ability to report information back to a central location-there are several interesting things you can do with that.

6. There will be an increase in three (potentially conflicting) areas [9]: deployment, commoditization, and sophistication of attack signature. For deployment, intrusion detection points will be showing up all through the network: at the network level (on the firewall, on the switch, on the router), at the system level (on servers, on desktops), and at the application level (on the database or SAP server, for example). Ever increasing speeds will push IDS technologies. For commoditization, given that there will be more and more detection points, they will become simpler to operate and more "appliance-like," so that they drop into the network infrastructure with no changes. It is thought as a "security toaster." For sophistication of attack signatures, the attack recognition logic will start to involve things like behavioral profiles and deviations from that profile (as an example). There will be much more "intelligence" in determining what constitutes an attack or resource misuse. Inevitably the total number of attacks will increase in concert with the increase in network speeds.

7. There will be merged IDS products that are burglar alarm and expert-based. Ideally, they will be merged into suites of network management tools.

8. The next issue to be addressed by commercial products is the inclusion of automated reactions to certain kinds of detected problems. This is beginning to emerge with systems changing firewall/router rules. There will be more of "active defenses" in the way detections and reactions are combined. This trend will increase over the next 2-4 years.

9. It is desirable to develop an IDS console, which will communicate with multiple pieces of the network architecture: firewalls, routers, switches and even different ID systems. Also, an IDS protocol or reporting format will be a design requirement: routers could relay SNMP traps and network statistics, firewalls could transfer failed packets for analysis and different ID systems could exchange findings. The possibilities are endless, leaving us with the definitive network monitor, manager and security package, thanks to the pooled efforts of each individual component.

In the next Section we define the research directions that could translate the above goals into realities.

4. DEFINING RESEARCH AREAS AND DIRECTIONS

Although intrusion detection is new to the commercial market, it's been around in the research labs for a couple of decades. Until recently, the commercial sector has paid little attention to it, and has to some extent been busy reinventing the wheel. But it is easy to predict that as the importance of security continues to increase, the gap between R&D and commercial products will shrink.

Among the most visible areas of active research [10] in the IDS community is the development of technologies to manage and interpret security relevant alert streams produced from an ever-increasing number of INFOSEC devices. In recent years, the growing number of security enforcement services, access logs, intrusion detection systems, authentication servers,

vulnerability scanners, and various operating system and applications logs have given administrators the leverage to have an insightful view into security-relevant activities occurring within their systems. The motivation for INFOSEC alarm correlation is straightforward: as we continue to incorporate and distribute advanced security services into our networks, we need the ability to understand the various forms of hostile and fault-related activity that our security services observe as they help to preserve the operational requirements of our systems.

Today, in the absence of significant field-able technology for security-incident correlation, there are several challenges [11] to meet while providing effective security management for mission-critical network environments:

- Domain expertise is not widely available that can interpret and isolate high threat operations within active and visible Internet-connected networks. Also not widely available are skills needed to understand the conditions under which one may merge INFOSEC alerts from different sources (e.g., merging firewall and OS syslogs with intrusion detection reports). In an environment where thousands (or tens of thousands) of INFOSEC alarms may be produced daily, it is important to understand redundancies in alert production that can simplify alert interpretation. Equally important are algorithms for prioritizing which security incidents pose the greatest administrative threats.

- The sheer volume of INFOSEC device alerts makes security management a time-consuming and therefore expensive effort. There are numerous examples of organizations that have found even small deployment of IDS sensors to be an overwhelming management cost. As a result, these IDS components are often tuned down to an extremely narrow and ad hoc selection of detection heuristics, effectively minimizing the coverage of the IDS tool.

- In managing INFOSEC devices, it is difficult to leverage potentially complementary information produce from heterogeneous INFOSEC devices. As a result, security relevant information that, for example, is captured in a firewall log, is typically manually analyzed in isolation from potentially relevant alert information captured by IDS, Syslog, or other INFOSEC alert source.

Additionally, other areas of research interests are beginning to emerge and most originated from the failures in the implementation of currently available IDS products. The following are the borderline issues at the center stage of these research works:

1. Data sets: Better data sets are necessary for better calculation of metrics in future evaluations and to further research. Datasets need to consist of many more examples of both attack and background traffic than have previously been available. Datasets need to be gathered collaboratively by a wide variety of researchers and stored centrally so that they represent a wide variety of network and system configurations and can be updated periodically without undue effort by any one entity. Datasets will need to take on new forms such as specifications and tools for created attack and background traffic in ones own environment so that IDS developers can explore use of new and different inputs for their systems.

2. Performance metrics: Metrics for IDS performance are a research topic in and of themselves, and will need to be expanded to better calculate and compare the amount by which an IDS improves the security of a given network configuration rather than simply tallying attack and false alarm rates.

3. Anomaly-based detection approach: Generally speaking, there seems to be much interest in going back to the anomaly-based approach of years ago without really understanding the value of what has been accomplished with the misuse detection approach. Thus, the industry is likely to move much faster to address the anomaly-based approach because of the successes and lessons learned from the misuse approach.

4. Expert-based approaches: A large number of IDS researchers are working on expert-based approaches because those are technically more interesting and are more likely to really evolve into something useful in the long run. The big gap is that the research tends to also ignore the "real security equals network management" problem and builds systems that are hard to manage, don't have intuitive user interfaces (or documentation) or that are cumbersome to use. It is likely that the good ideas from the R&D systems will wind up in commercial products. This will be the right research direction since good ideas, not products, come from research.

5. Data correlation: It could be successfully argued that the future of IDS lies in data correlation research. The IDS of tomorrow will produce results by examining input from several different sources. The notion of NIDS and HIDS will disappear, leaving us with a group of distributed components performing specific tasks.

The concept of HIDS plays an important role in this scenario. Encrypted traffic demands that we shift packet analysis, an important part of ID, to the host. It is difficult to find another solution. There is however, a distinct advantage gained by using this method of analysis. The signatures can be tailored to one host, as opposed to the heterogeneous mix of Microsoft, Unix and application specific rules in place on most NIDS. Moreover, a recurring scan could quickly monitor which services/programs run on a machine, allowing for an even more precise rule-set. So, instead of a sensor capturing all traffic on a network, the client machines will monitor their own traffic.

Research efforts will vigorously continue in this area as the way to solve this problem lies in statistical analysis and

predictive artificial intelligence performed on strange data sets. The management console, having received these abnormal event notifications from many clients, needs to concentrate on possible relationships, relevance and correlation. It needs to determine the likely triggers for such infrequent events. Moreover, the console will need to communicate with multiple pieces of the network architecture: firewalls, routers, switches and even different IDSs.

6. Audit trails: Research to determine what kinds of information should be in audit trails, and when such data needs to be collected to optimally drive any intrusion detection system will be critical in defining the architecture of data mining technologies.

7. Storage format: Research to determine the best structure/storage formats for audit data so that it can be quickly processed without taking up huge amounts of storage will aid data mining architectural designs.

8. Software automation: Exploring how to define policy in a consistent and meaningful way such that it can be expressed in software for automated comparison and detection of intrusions and internal misuse is a viable research field.

9. Reference model: There is a need to develop a reference model for IDS design as any meaningful design should take a queue from a standard reference model just as the one done by Christopher Schuba on a formal reference model for firewalls. The lack of one was prevalent in the early days of firewall development when people were so busy building firewalls, selling firewalls, giving firewall tutorials, and hyping firewalls that they neglected to study what really should constitute a firewall.

Something similar would benefit research and development intrusion detection systems. There is so much need, and so much pressure will soon be applied from various quarters, that basic scientific foundational research won't be done. Already, the focus is largely on engineering research-how to build a better version without understanding the underlying principles. The security marketplace/government has a poor track record of helping support research in academia, while at the same time offering incredible incentives to lure away promising students and even some faculty who are best suited to do this work.

Thus, there is the need to have the commercial realm-both vendors and customers-providing support to academia to do basic research rather than simply implementing small variations of the same ideas all over again (which is what is going on at many places right now). This is an area where there is a need for some radically new ideas investigated. Some well-targeted research could pay off in the long run with better technology.

5. CONCLUSION

The issues discussed in the preceding Sections makes us to believe that the IDS are here to stay, although future systems will undoubtedly take a different form than current versions. The ideas presented here, while optimistic, are attainable. Presently, there are several ongoing research and development efforts worldwide. For example, the mathematical and artificial intelligence concepts required for success in new technologies are already being developed, tested and improved upon. A USA company - SRI has a great start with the NIDES and EMERALD projects, using a distributed model. Equally, Internet Security Systems, Inc. (ISS) [12] is developing products that will scan networks for vulnerabilities and modify the IDS filters based on the results. So also is Lancope's StealthWatch [13] that is using a " flow-based architecture to recognize abnormal behavior. Several other refinements and features outlined above are being incorporated into upcoming products, all of which will improve with time and research.

Finally, as security continues to move to the center stage, and with vendors and other research organizations working to eliminate the shortcomings of IDS products, the future looks brighter for this technology. Future IDS will merge all of the independent network components and tools, which exist today, into a complete and cooperative system, dedicated to keeping networks, stable. There will be many distributed elements performing specific jobs, each passing the results onto a higher level for correlation and analysis.

REFERENCES

[1] Iheagwara C. and Blyth A, " Evaluation of the performance of IDS systems in a switched and distributed environment," Computer Networks, 39 (2002) 93-112

[2] Cohen F, Simulation Cyber Attacks, Defenses, and Consequences, Computer & Security, p479-518, 18, 1999.

[3] Richards K, "Network Based Intrusion Detection: a review of technologies," Computers & Security, 18 (1999) 671-682.

[4] Iheagwara C., Blyth A., Singhal M., "A Comparative Experimental Evaluation Study of Intrusion Detection System Performance in a Gigabit Environment," Journal of Computer Security, Vol 11(1), January, 2003

[5] Lee W. et. al, "Toward Cost-Sensitive Modeling for Intrusion Detection and Response," North Carolina State University, 1999.

[6] Irvine C. et al " Toward a Taxonomy and Costing Method for Security Metrics," In: Proceedings of the Annual Computer Security Applications conference, Phoenix, AZ, Dec. 1999

[7] State of the Practice of IDS Technologies http://www.sei.cmu.edu/publications/documents/99.reports/99tr028/99tr028exsum.html

[8] http://www.silicondefense.com/spice

[9] Limitations of Network Intrusion Detection by Steve Schupp
http://www.sans.org/infosecFAQ/intrusion/net_id.htm

[10] Porras P. A. " Intrusion Report Correlation" , 18th Annual Computer Security Applications Conference Las Vegas, Nevada, Dec. 9-13, 2002
[11] Protect your network with an Intrusion Detection system, Gartner Research http://www.techrepublic.com/article.jhtml?src=search&id=r00520010209ggr01.htm
[12] http://www.iss.net
[13] http://www.StealthWatch/products/index.htm

Appendix 12

"Future Directions in the Development of Intrusion Detection Systems" The Information Systems Control Newsletter, May 2003

POST AUDIT REVIEW

Published by ISACA's National Capital Area Chapter

May 2003

President's Message

Ben Hsiao, President

Our chapter won the K Wayne Snipes award this year as the best ISACA chapter worldwide.

Dear Fellow Members:

The 2002-2003 Chapter year is coming to a close. In my last message to you as President, I wanted to express my thanks to all of the volunteer board members for their consistent dedication and cooperation. The results we achieved through our voluntary efforts-cost effective and meaningful special and luncheon seminars, informative newsletters and web sites, and continued positive financial posture, have not gone unnoticed. Our chapter won the K Wayne Snipes award this year as the best ISACA chapter worldwide. Jay Jacobsen, the in-coming president for 2003-2004, and I will be attending the annual ISACA leadership conference in May at Houston to accept the award on behalf of our chapter.

Although we should all be proud of the international recognition our chapter received, we cannot lessen our drive or effort to improve the quality of our services for the chapter members. To remain a consistent top-notch chapter, we need continued infusion of dedicated volunteers to bring in fresh ideas on the type and delivery of services for our chapter members. Over the last couple of years, we have dramatically improved our services including allowing the use of credit card to pay for seminars, adding incentives to increase seminar attendance, and using an on-line service to conduct the annual survey of membership needs. Another example is the recently approved change to distribute future chapter newsletter and other special announcements via email and/or the Web. I hope many of you consider volunteering to serve on the board.

As in my previous messages, I want to recognize the volunteer service of one of the Board members. Hanh Do, serving on the CISA Review and Certification Committee, has been the person spending considerable effort in setting up the annual quality CISA review class for the past three years. She ordered the study material in advance and coordinated with the instructors to schedule each review class. She registered the attendees and made sure they received the study materials. She also attended the 5 Saturday sessions to resolve last minute difficulties and provided valuable feedbacks on the instructors and the materials provided. Because of Hanh's dedication, the CISA review classes has always been well attended and with favorable feedback.

Hanh is a dedicated volunteer working very diligently with the rest of the board members to keep the cost reasonable and still provide chapter members with a top quality CISA review class. We on the Board intend to continue being responsive to your needs. As always, we do want constructive feedback from you. Please contact any board members by phone, email, or through our Website.

I hope you will continue to take advantage of the services provided to you by our chapter. Best wishes for the coming ISACA-NCAC year.

Ben Hsiao

2003 Chapter Luncheon Meetings held at the Holiday Inn Capitol, 550 C Street, SW (Located 1 Block from L'Enfant Plaza Metro Station) 5/27/03 and 6/24/03

Future Directions in the Development of Intrusion Detection Systems

ıthor: Charles Iheagwara and ıdrew Blyth

ıO Introduction

ı Intrusion detection system (IDS) is ıecurity system that monitors ımputer systems and network traffic ıd analyzes that traffic for possible ıstile attacks originating from outside ı organization and also for system ısuse or attacks originating from ıside the organization. Historically, ıtrusion Detection Systems (IDS) ılved from the system audit informa- ın analysis [1]. A strong need for IDS ıerged due to the fact that system dit information, for example system ; files, were difficult to decipher, ılyze, and review by auditors and ıtems administrators.

ı IDS technology is very powerful. It ıvides an incredible view into the ırity problems facing an organiza- ı. However, it still requires consid- ble refinements to eliminate the ıknesses in currently available ducts. Some of the weaknesses that considered short-term i.e. lability, hierarchical reporting, and amic remote updates are already ıg addressed by vendors and some ıe solutions are in the alpha- or ı-testing stage. The long-term knesses, which by and large are ıics for IDS performance, are being ıessed as research topics in and of ıselves.

products, much like the security stry itself, have grown rapidly over ıast few years. These tools have me essential security components - luable as a firewall to many ıizations . However, as in any environment, things change. As networks and crackers evolve and grow rapidly, demanding that security tools keep up, the IDS faces several daunting but exciting challenges in the future and are sure to remain one of the best weapons in the arena of network security.

Among the many challenges is the development of commercial IDS products to meet the needs of today's complex networked environments. To effectively deploy the IDS in such complex and networked environments requires a broad understanding of computer security and good product delivery. As the information technology landscapes and infrastructures become more and more complex so also has the performance effectiveness of the IDS diminished in such environments [2]. The complexities have often resulted into the most significant obstacle to the success of an information security improvement initiative due to lack of management support. In surveys conducted by security trade magazines [3], lack of management support was cited as one of the principal barriers to effective information security.

Within the IDS market place are two broad categories of product: Host-based and Network-based. Generally, the commercially available IDS products shown in Table 1 are classified according to their approach to intrusion detection, all being either host or network-based. None of the products integrate host-based and network-based intrusion detection capabilities and a few integrate security assessment capabilities with basic IDS functionality, such as audit trail analysis and malicious software protection.

See Table 1 on next page.

Each implementation has been plagued with a couple of problems some of which are discussed in Section 2.

The solution to some of the problems could come from the insights and operational experience gained from the implementation of currently available IDS products. In this regard, results of previous research studies on performance effectiveness could be helpful in evolving an effective design approach and charting the course for new products development. Crucial to this is the results of research studies that established a link between IDS performance in technical, operational and cost/economic terms and product architectures.

With so many studies on IDS technologies, those that are performance-based have leapt forward to take center stage as developers race to refine existing products. IDS performance studies [2,4,5] measure the effectiveness of the IDS using a variety of metrics for different environments, as described below.

Richards [4] evaluates the functional and performance capabilities of the industries' leading commercial type IDS. In the areas tested, the performance of the IDS was rated based on their distinctive features, which were characterized into different performance indexes. The research work represented a new direction for ID systems in that it moved the focus away from scientific concepts research to performance evaluation of the industries' best products. However, the study was limited to a small proto

Vendor	Product Name	Approach to ID	Platform
ISS	RealSecure	Network-based Packet capture, signature analysis, and real-time playback	UNIX and NT
Cisco (formerly WheelGroup)	NetRanger	Network-based. Passive network monitor with packet filtering router	UNIX
Security Dynamics (formerly Intrustion Detection)	Kane Security Monitor	Host-based. Passive ID capabilities with assessment functions	NT
DMW Worldwide	HostCHECK	Host-based. Passive ID capabilities (audit trail analysis and file checksums) and assessment functions	UNIX
MEMCO (formerly AbirNet Ltd.)	SessionWall	Network-based, Real-time connection and playback	NT

Table 1: IDS products

design isolated and non-switched network which did not reveal the impact of packet switching on the accuracy and ability to capture attack packets in their entirety. Iheagwara et al. [2] expands this effort with an evaluation study of the effect of deployment techniques on IDS performance in switched and distributed system. The study demonstrates that monitoring techniques could play an important role in determining the effectiveness of the IDS in a switched and distributed network.

In an experimental evaluation study of intrusion detection system performance in a gigabit environment, Iheagwara et al. [5] examines the system benefits of using a single Gigabit IDS sensor instead of multiple Megabit sensors for a wide range of defined system attacks, network traffic characteristics, and for their contexts of operational concepts and deployment techniques. The study established the relationship between traffic parametric values and the IDS performance for specified environments.

The results of cost-benefit models and analysis studies of IDS deployments, although relatively few sheds some light on the integral economic building blocks that could serve as a guide for acquisition and deployment of the products. Lee et al. [6] study the problem of building cost-sensitive intrusion detection models. For intrusion detection, Irvine [7] defines auditing of network control functions in intermediate nodes, and rule-based network intrusion systems in the total subnet as the mechanisms. Irvine also discusses the costs of those security services and mechanisms. A design cost component model, which could serve as a guide for new product development is yet to be developed.

While these studies characterize areas of design and implementation weaknesses, none have provided IDS developers with the data and techniques necessary to create truly "next-generation" intrusion detection algorithms and tools. For instance, none of the performance evaluation studies [2,4,5,] that explore the relationship between deployment techniques and attack system variables, and the performance of the IDS; and the studies [6, 7] on cost models that investigate the cost-benefit/sensitive analysis for intrusion detection deployment presents developers with any concrete guide on how to translate research results into practical design tools.

Going by the results of the studies, the need to establish a uniform framework for the new products development or the refinement of existing ones using proven results of research studies and the operational experiences gained from IDS implementations cannot be overstated.

Therefore, this paper will seek to chart a course for the future development of new IDS products by drawing from the field experiences of the authors in the operation of the IDS technology. Further, we will demonstrate that the concepts proposed here could reasonably be associated with well thought out propositions on the requirements and functions that must be satisfied in order for new products to be implemented successfully from design to lifecycle management.

2.0 Problems with the current designs

There are pitfalls in the current implementation of commercially available IDSs. The pitfalls include the issues of variant signatures, lag time between the time an attack is discovered and an IDS signature is provided to users (catch-up), false positives and negatives alerts, data overload, difficul-

ts to function effectively in switched vironments and scalability issues.

riants. While the ability to develop d use signatures to detect attacks is a eful and viable approach, there are ortfalls to using this approach as gnatures are developed in response to w vulnerabilities or exploits that have en posted or released. However for a gnature to be effective, it must be fficiently unique to only alert on alicious traffic and not infringe on alid network traffic. The difficulty re is that exploit code can often be sily changed. It is common for an ploit tool to be released and then ve its defaults changed shortly reafter by the hacker community.

Catch-up. New signatures can only be veloped once an attack has been dentified. Therefore between the eation of an attack and the deploy- ent of a signature to detect the attack, window of opportunity exists for an truder to mount an attack with little to chance of the attack being detected.

False positives. A common complaint is he amount of false positives an IDS enerates. Developing unique signa- ures is a difficult task and often times he vendors will err on the side of lerting too often rather than not nough. This is analogous to the story f the boy who cried wolf. It is much ore difficult to pick out a valid trusion attempt if a signature also lerts regularly on valid network tivity. A difficult problem that arises how much can be filtered out without otentially missing an attack.

alse negatives. Detecting attacks for hich there are no known signatures. his leads to the concept of false gatives where an IDS does not nerate an alert when an intrusion is tually taking place. Simply put, if a gnature has not been written for a rticular exploit, there is an extremely good chance that the IDS will not detect it.

Data overload. Another aspect, which does not relate directly to misused detection but is extremely important is how much data can be analyzed effectively and efficiently. Depending on the intrusion detection tools deployed by a company and its size, there is the possibility for logs to reach millions of records per day.

Difficulties in switched environments. Network capture and analysis in a switched LAN environment usually means "tapping" the switch's lines by using a "mirror" port or deployment in other tapping configurations. In this approach, traffic is copied from one "source" port to another destination or "mirror" port. Mirroring a full duplex source port may cause packet loss as traffic on the full duplex source port exceeds the available bandwidth of the mirror port.

Scaling up:

In the last couple of years, there has been a significant increase in network traffic utilization. With this has come the introduction of Gigabit Ethernet technology to accommodate this increase in bandwidth – and thus the volume of traffic to be analyzed. The problem associated with this is that older IDS technologies that operate at 10mbps or 100mbps bandwidths are overwhelmed with the increase in traffic volume. With Gigabit Ethernet, the older IDS technologies become seriously overloaded.

3.0 Future design directions and improvements

Current IDS products bring the ability to view network and system activity in real-time, identify unauthorized activity and provide a near-real-time automated response. IDS products also provide the ability to analyze today's activity in view of yesterday's activity to identify larger trends and problems. It is reasonable to expect IDS technology to revolutionize computer security efforts, by allowing real-time operational capability in controlling unauthorized activity in corporate cyberspace. IDS technology does not directly address other security issues such as identification/authentication, confidentiality, etc., although some of these technologies will be integrated with IDS in the near future.

Generally, expectations typically revolve around accuracy and the myth of the "silver bullet" (the latter is something that all security products have to face). Realistic expectations are that intrusion detection systems are discovery and detection tools that guide further investigation. Unrealistic expectations are that intrusion detection systems, like firewalls, will automatically protect all users from all threats. The following are expectations of future IDS products.

1. An IDS deployment should have some operational procedure behind it to gather additional information and fine-tune the network and the process. A good IDS will automate as much of this process as possible. Many customers think that a given security product like an IDS will protect them from 100% of the "bad things." In a practical world, there are no absolutes, instead an IDS can significantly reduce the risk from network-based threats, but they are not perfect.
2. The IDS product should detect, in near real-time, any kinds of attempts to exploit known weaknesses, or to probe your internal network. They should also keep track of attempts to overload necessary resources. Along with this, they should perhaps sound an alarm, trigger some predefined action, and keep a good log for analysis.

3. A reasonable expectation is to have such a system detect attacks against previously unknown vulnerabilities, detect slow attacks by insiders, have built-in automatic protection for your network when an attack is detected, or to be operable in "hands-off" mode.

4. Because of the uncertain nature of security policy and methods for detecting violations, any current or near-future system that is likely to be able to detect intrusions and misuse is also going to generate false alarms. Someone with enough knowledge of the environment and the nature of the ID system to sift through alarms will be required to decide which ones are false alarms (mistakes, bugs, harmless curiosity), and which are real attacks.

   Single-sensor IDS performance has been taken about as far as possible, with databases of attack signatures and methods to update these databases as the only areas of improvement. . The next big step is distributed intrusion detection, which takes the results from multiple sensors, deployed throughout the enterprise, and correlates them into a single "big picture" view of the network. Manual correlation is the only way to do this currently. . However there will be several "software assist" programs marketed in the next 18-24 months, similar to what is currently happening in the firewall market. There will be several products; all more or less interchangeable at their core, with differentiation based on the "assorted bells and whistles" bundled with the base. One area to pay close attention to is what the vendors do to take advantage of all these sensors deployed throughout the network, all with the ability to report information back to a central location. (what interesting things are you talking about? slj)

5. We expect that there will be an increase in IDS functionality in three (potentially conflicting) areas: deployment, commoditization, and sophistication of attack signature. Concerning deployment, intrusion detection points will be available throughout the network environment: at the network level (on the firewall, on the switch, on the router), at the system level (on servers, on desktops), and at the application level (on the database or SAP server, for example). Concerning commoditization, given that there will be more and more detection points, they will become simpler to operate and more "appliance-like," so that they drop into the network infrastructure with no changes, i.e. a "security toaster." Concerning sophistication of attack signatures, the attack recognition logic will involve items such as behavioral profiles and deviations from that profile (as an example). There will be much more "intelligence" in determining what constitutes an attack or resource misuse. Inevitably the total number of attacks will increase in concert with the increase in network speeds.

6. We expect that future IDS products will merge the burglar alarm and expert-based into suites of network management tools.

7. The next issue is the inclusion of automated reactions to certain kinds of detected problems into commercial products. This is beginning to emerge with other systems changing firewall/router rules, such as "active defenses"; the way detections and reactions are combined. This is likely to become more widespread over the next 2-3 years.

4.0 CONCLUSION

The current architecture of commercially available IDS products is built primarily out of the perceived role of intrusion detection. It is equally true that due to the complexities in evolving a uniform IDS technology, the current implementation is far from achieving the desired intrusion detection goals. Thus, the architecture and complexities in the current IDS technologies have given rise to obvious operational and deployment difficulties. However, as a result of current research and development efforts it is expected that the next generation of intrusion detection systems will not inherit the current pitfalls.

A robust future IDS product should be designed with the capabilities to generate alarms, display alarms, clear alarms, and provide context-sensitive on-line help. It will also have a database mechanism and sophisticated built-in reporting for effective data management. These tools will allow a security management staff to analyze the data as desired. Ultimately when this is realized, IDS products will serve the IT Security Community more efficiently and cost effectively.

Finally, it is expected that the future IDS product will be a grocery shelf of choices that will fit a broad range of needs. It won't be a single product, but an integrated system of products from multiple vendors. It will include network-based sensors, host-based sensors, and a centralized anomaly detection system that analyzes logs sent to it by the sensors. The anomaly detection system will take predetermined actions depending on the nature and severity of the detected threat.

References

[1] "The Evolution of Intrusion Detection Systems" http://www.securityfocus.com/infocus/1514

[2] Iheagwara C. and Blyth A, "Evaluation of the performance of IDS systems in a switched and distributed environment," *Computer Networks*, 39 (2002) 93-112

[3] Cohen F, Simulation Cyber Attacks, Defenses, and Consequences, Computer & Security, p479-518, 18, 1999.

[4] Richards K, "Network Based Intrusion Detection: a review of technologies," *Computers & Security*, 18 (1999) 671-682.

[5] Iheagwara C., Blyth A., Singhal M., "A Comparative Experimental Evaluation Study of Intrusion Detection System Performance in a Gigabit Environment," *Journal of Computer Security*, Vol 11(1), January, 2003

[6] Lee W. et al, "Toward Cost-Sensitive Modeling for Intrusion Detection and Response," North Carolina State University, 1999.

[7] Irvine C. et al "Toward a Taxonomy and Costing Method for Security Metrics," In *Proceedings of the Annual Computer Security Applications conference*, Phoenix, AZ, Dec. 1999