

The Effective Combating of Intrusion Attacks through Fuzzy Logic and Neural Networks

By

Robert Melvin Goss

The Effective Combating of Intrusion Attacks through Fuzzy Logic and Neural Networks

by

Robert Melvin Goss

Dissertation

Submitted in fulfilment

of the requirements

for the degree

Magister Technologiae

in

Information Technology

in the

Faculty of Engineering, the Built Environment

and Information Technology

of the

Nelson Mandela Metropolitan University

Supervisor: Prof R. von Solms

Co-Supervisor: Dr M. Botha

January 2007

Declaration

I, Robert Melvin Goss, hereby declare that:

- The work in this dissertation is my own work.
- All sources used or referred to have been documented and recognized.
- This dissertation has not previously been submitted in full or partial fulfilment of the requirements for an equivalent or higher qualification at any other recognized educational institute.

Robert Melvin Goss

Abstract

The importance of properly securing an organization's information and computing resources has become paramount in modern business. Since the advent of the Internet, securing this organizational information has become increasingly difficult. Organizations deploy many security mechanisms in the protection of their data, intrusion detection systems in particular have an increasingly valuable role to play, and as networks grow, administrators need better ways to monitor their systems. Currently, many intrusion detection systems lack the means to accurately monitor and report on wireless segments within the corporate network. This dissertation proposes an extension to the NeGPAIM model, known as NeGPAIM-W, which allows for the accurate detection of attacks originating on wireless network segments.

The NeGPAIM-W model is able to detect both wired and wireless based attacks, and with the extensions to the original model mentioned previously, also provide for correlation of intrusion attacks sourced on both wired and wireless network segments. This provides for a holistic detection strategy for an organization. This has been accomplished with the use of Fuzzy logic and neural networks utilized in the detection of attacks. The model works on the assumption that each user has, and leaves, a unique footprint on a computer system. Thus, all intrusive behaviour on the system and networks which support it, can be traced back to the user account which was used to perform the intrusive behaviour.

Acknowledgements

It is a pleasure to thank the following people for both their help and support over the past two years whilst performing this research.

- I would like to first and foremost like to thank the Lord Jesus Christ, without whom, this work would not be possible.
- Melvin and Julia Goss, my loving parents who taught me that all things are possible if you put your heart and mind into it.
- My brother and sister, for their endless encouragement.
- My supervisor Prof Rossouw von Solms, for his supervision and support.
- My co-supervisor Dr. Martin Botha, for his enthusiastic supervision, guidance and support.
- My colleagues at the Nelson Mandela Metropolitan University (NMMU), for their assistance and friendship.
- Finally I would like to thank the National Research Foundation (NRF) for their financial assistance.

Contents

DECLARATION	I
ABSTRACT	III
ACKNOWLEDGEMENTS	IV
CONTENTS	VI
LIST OF FIGURES.....	IX
LIST OF TABLES	X
CHAPTER 1	1
INTRODUCTION	1
1.1 <i>Motivation for this Study</i>	4
1.2 <i>Problem Statement</i>	6
1.3 <i>Objective</i>	6
1.4 <i>Methodology</i>	7
1.5 <i>Layout of the Dissertation</i>	8
1.6 <i>Conclusion</i>	11
CHAPTER 2	12
INTRUSION DETECTION IN GENERAL	12
2.1 <i>Introduction</i>	12
2.2 <i>Computer Security Concepts in General</i>	13
2.3 <i>Real-World Computer Security Problems</i>	20
2.4 <i>Protection Mechanisms</i>	22
2.5 <i>Intrusion and Intrusion Detection Concepts</i>	24
2.5.1 Misuse detection in general.....	25
2.5.2 Anomaly Detection in General.....	26
2.6 <i>History of Intrusion Detection Systems</i>	27
2.7 <i>Conclusion</i>	28
CHAPTER 3	30
IMPORTANT NETWORK STANDARDS AND ATTACKS.....	30
3.1 <i>Introduction</i>	30
3.2 <i>Networking in General</i>	31
3.2.1 History of Networking.....	31
3.2.2 Modern Networks.....	32
3.2.3 The Future of Networks	34
3.3 <i>OSI Reference Model</i>	36
3.4 <i>802.3 Ethernet Standard</i>	38

3.5	<i>802.11 a, b, g and n Wireless Standards</i>	39
3.6	<i>802.16 WiMAX Wireless Standard</i>	41
3.7	<i>Hacking Wireless Networks</i>	43
3.8	<i>Generic Hacks and Attacks</i>	45
3.9	<i>Conclusion</i>	51
CHAPTER 4		52
WIRELESS INTRUSION DETECTION TAXONOMY AND PRODUCTS		52
4.1	<i>Introduction</i>	52
4.2	<i>Intrusion Taxonomies</i>	53
4.2.1	Bishop's Vulnerability Taxonomy.....	54
4.2.2	Aslam's Taxonomy	55
4.2.3	Neumann & Parker's Taxonomy	56
4.2.4	Lindqvist & Jonsson's Intrusion Taxonomy	58
4.2.5	Landwehr's Taxonomy	59
4.3	<i>Proactive Generic Intrusion Taxonomy</i>	60
4.4	<i>Commercial, Research and Public Domain IDSs</i>	64
4.4.1	Public Domain IDSs	65
4.4.2	Commercial IDSs	66
4.4.3	Research IDSs.....	70
4.5	<i>Limitations of Current Intrusion Detection Systems</i>	72
4.6	<i>Characteristics of Wireless Intrusion Detection Systems</i>	76
4.7	<i>Conclusion</i>	79
CHAPTER 5		80
THE UPDATED PROACTIVE IDENTIFICATION MODEL (NEGPAIM-W)		80
5.1	<i>Introduction</i>	80
5.2	<i>Design Specifications for Updated Model</i>	81
5.3	<i>Hierarchical Hybrid Architecture</i>	83
5.4	<i>The Conceptual Model: The Components</i>	87
5.5	<i>The Model in Perspective</i>	91
5.6	<i>Conclusion</i>	93
CHAPTER 6		94
THE FUZZY, NEURAL AND CENTRAL ANALYSIS ENGINES.....		94
6.1	<i>Introduction</i>	94
6.2	<i>The Fuzzy Engine</i>	94
6.2.1	Alternative approach to misuse detection.....	95
6.2.2	Fuzzy methodology	109
6.2.3	Dynamic proactive identification model	113
6.3	<i>The Neural Engine</i>	116
6.3.1	Anomaly Detection Through Neural Networks	117
6.3.2	User Identification Strategy	120
6.4	<i>The Central Analysis Engine (CAE)</i>	121
6.4.1	Functions of the CAE	122
6.4.2	The central analysis methodology	123
6.4.2a	System Activities	126

6.4.2b	Administrator's Activities	128
6.5	<i>Detection Example</i>	130
6.6	<i>Conclusion</i>	133
CHAPTER 7		135
A WIRELESS INTRUSION ATTACK EXPERIMENT		135
7.1	<i>Introduction</i>	135
7.2	<i>The NeGPAIM-W Prototype</i>	136
7.2.1	Sentinel IDS	136
7.2.2	Implementation of Sentinel IDS	137
7.2.3	Implementation of the Fuzzy Engine	140
7.2.4	Fuzzy Engine Configuration	142
7.2.5	Implementation of the Neural Engine	144
7.2.6	Neural Engine Configuration	146
7.2.7	Implementation of the Central Analysis Engine	149
7.3	<i>Prelude to the Experiment</i>	150
7.3.1	Experiment Environment	151
7.3.2	Intrusion Tools	151
7.4	<i>The Experiment</i>	153
7.5	<i>The Results</i>	157
7.5.1	The Fuzzy Engine's Response	158
7.5.2	The Neural Engine's Response	160
7.5.3	The Central Analysis Engine's (CAE) Response	161
7.6	<i>Conclusion</i>	162
CHAPTER 8		163
THE CONCLUSION		163
8.1	<i>Introduction</i>	163
8.2	<i>Review of the Problem Statement</i>	164
8.3	<i>Meeting the Dissertation Objectives</i>	166
8.4	<i>Further Research</i>	168
8.5	<i>Conclusion</i>	169
ANNEXURE A		171
UTILIZING FUZZY LOGIC AND NEURAL NETWORKS FOR EFFECTIVE, PREVENTATIVE INTRUSION DETECTION IN A WIRELESS ENVIRONMENT		171
BIBLIOGRAPHY		184

List of Figures

Figure 1.1: Proposed layout of dissertation	10
Figure 3.1: OSI Reference Model.....	36
Figure 3.2: Simple WiMAX Implementation.....	42
Figure 3.3: Man-in-the-Middle Attack.....	47
Figure 3.4: Replay Attack.....	48
Figure 4.1: Lindqvist & Jonsson's Taxonomy.....	59
Figure 4.2: Proactive Generic Intrusion Taxonomy (Botha, 2003)	61
Figure 4.3: Updated Proactive Generic Intrusion Taxonomy	64
Figure 5.1: Updated Model Design Specifications.....	82
Figure 5.2: Hierarchical Architecture (Botha, 2003).....	86
Figure 5.3: General Representation of NeGPAIM-W.....	87
Figure 6.1: Comparison between Traditional and Alternative Misuse Detection	96
Figure 6.2: Unauthorized Client Probe Request (Interlink Networks, 2002).....	102
Figure 6.3: Alternative Misuse Detection Approach (Wired)	106
Figure 6.4: Alternative Misuse Detection Approach (Wireless)	107
Figure 6.5: Sample Fuzzy Graphs.....	111
Figure 6.6: FCM for Initial Access Phase of a Wireless Attack.....	115
Figure 6.7: Relationship Diamond Model.....	119
Figure 6.8: Example of Calculation for Gaining Total Intrusion Probability (Botha, 2003).....	123
Figure 6.9: Flow Chart of Central Analysis Methodology (Botha, 2003)	125
Figure 7.1: Sentinel IDS Layout.....	138
Figure 7.2: Low-Level Detection Model (Botha, 2003).....	139
Figure 7.3: Fuzzy Engine Components Diagram	141
Figure 7.4: Example of a Fuzzy Sensor Questionnaire.....	143
Figure 7.5: Example of a Fuzzy Sensor Template Graph	144
Figure 7.6: Neural Network Training (Initialization)	147
Figure 7.7: Neural Network Training (Footprinting).....	148
Figure 7.8: Neural Network Training (Completed)	149
Figure 7.9: Setup of the Experiments.	154
Figure 7.10: Example Sniffing of Wireless Traffic.	154
Figure 7.11: Example Cracking AP1 WEP Key.	155
Figure 7.12: Example Portscan on Host1.....	156
Figure 7.13: Example Bruteforce Attack on Host1.	157

List of Tables

Table 2.1: Security Control Classes and Families (Katzke et al., 2005; Goguen et al., 2002)17

Table 4.1: SRI Neumann & Parker (NP) Taxonomy.....57

Table 4.2: Summary of IDS’s properties71

Table 6.1: Weighting Structure for Wired Network Non-Linear Function (Botha, 2003).....103

Table 6.2: Weighting Structure for Wireless Network Non-Linear Function104

Table 7.1: Total User Behaviour String146

Chapter 1

Introduction

Since the inception of computers, they have become an integral and indispensable part of both organizations and normal people's lives. Information Communications Technology (ICT) has been steadily advancing at a rapid pace (LT Consultants & Buck Consultants, 2002). The Internet, which is one of the main implementations of ICT, is increasing more rapidly than that of any other communication technology in the 20th century. In 2000, it was estimated that approximately one half of US households were online (Wilhelm, 2000). The number of Internet hosts online has also been steadily increasing from 44 million in January 1999, to 88 million in August 2000, to almost 120 million in April 2001 (Telcordia Technologies, 2001). Web commerce has become the mainstream, with millions of people utilizing it to do online purchasing and to do business and private banking yearly. New consumers are now on the scene, buying items online instead of in a regular store. These consumers are known as cyber consumers (LT Consultants et al, 2002).

Wireless data technologies, such as Wi-Fi and High-Speed Downlink Packet Access (HSDPA), are the latest buzzwords in communications technology, with over ten million households in the USA alone having Wi-Fi based access-points installed, providing an Internet connection, according to Schiesel (2005). With business being conducted over the Internet and wireless private and public networks, businesses have had to make information available to individuals outside of their organization (DeYoung et al, 2002). This availability of information has led to security holes also becoming available to the public facing Web servers on the business's network.

With the advantages provided by public networks, such as the Internet and private and public wireless networks, organizations have not only become dependent on these networks, but also have a need for security to protect valuable data on their networks from malicious users outside their organizations. Nowadays, it has also become more and more necessary to protect networks against disgruntled inside users. Organisations must, therefore, protect their networks from corruption, theft or disruption of the flow of their personal and business data. The need for proper protection of an organization's information is becoming more and more important every year. This can be seen in the Annual CSI/FBI Cyber Crime Survey, which estimates that the amount of money lost by companies per annum due to attacks, system breaches and theft of information is around the figure of \$141,496,560. These losses are down from 2003, when the total losses for the year were estimated at \$201,797,340 (Gordon et al. 2004).

There are many ways to protect valuable data, e.g., Firewalls, Antiviruses, Access Controls, Policies and Intrusion Detection Systems. Security measures, such as access control and policies, while important in a security framework, often are not enough to stop attacks on a network. Attacks, such as Denial of Service (DoS), Distributed Denial of Service (DDoS), SMURF (a flood of ICMP echo requests) and Distributed Reflective Denial of Service (DRDoS), to name a few, work by flooding a network with traffic; therefore, denying services to the users on the network (CERT, 2005). Attacks of this nature often thwart access control and policies. The only way to stop attacks of this kind is to actively disable either the port on which they are attacking or, in extreme cases, to disable the connection to the Internet on the edge routers, until the source of the attack can be determined and blocked (Cisco Systems, 2003a). This, unfortunately, not only thwarts the attack, but also stops the flow of business data, often resulting in high financial loss.

A far more effective way to solve the last-mentioned problem is to implement and maintain an Intrusion Detection System (IDS). Intrusion Detection (ID) is defined as the art of identifying inappropriate, incorrect, or anomalous activity (Esposito et al., 2004). There are three main types of Intrusion Detection Systems, namely HIDSs (Host-Based IDSs), NIDSs (Network-Based IDSs) and Hybrid IDSs, which combines the two afore-mentioned types to form a more rounded IDS. HIDSs reside on a single host and usually monitor and protect the system

configuration and files from abnormal changes. Files and system settings are given weightings and an administrator can be alerted at any suspicious activity. In NIDSs the IDS monitors multiple nodes on the network and detects attacks by searching network traffic for patterns of known attacks and anomalous activity known from previous baselines. This traffic could have a source external to the organization, but have an IP address of a machine internal to the network. This is known as IP spoofing and NIDSs can be setup to identify this kind of attack. Both NIDS and HIDS require a database of previous known attacks to detect most attacks (Lehmann, 2005; Whitman & Mattford, 2003).

IDSs can provide an organization with protection for its valued data, be it client, product or sales information. An IDS provides this protection in a way that other protection mechanisms, such as encryption, firewalls, cipher locks and access control cannot. Most of the previously mentioned protection mechanisms have been designed to deny access, but an IDS has been designed to detect misuse or anomalous activity and report on it and, in some cases, stop the intrusion (CTA, 2002). This means that an IDS not only detects the attack, but also alerts the administrator to these intrusive events, providing an audit trail back to the attacker. IDSs provide an advantage by reading the logs of application and services, including the logs of other protection mechanisms. This ability to gain a holistic view of what is currently occurring on the system is what gives an organization an edge when an intrusion attack takes place. An IDS also allows the administrator to draw reports of intrusion events on his/her network, allowing him/her greater control over his/her network.

Although IDSs may sound like a miracle cure to the intrusion attacks that occur on organizational networks, this is simply untrue. IDSs do have a few problems associated with them, one of which is the inability to address the high volume of traffic across modern networks (McAfee, 2003). This problem is further compounded as network technologies, such as 10Gigabit networks, become main-stream. Another serious problem currently plaguing IDSs is the inability to detect attacks over wireless network media, including the correlation of wireless attacks with other wired, network-based attack data (Innella, 2002).

This dissertation focuses on the field of Intrusion Detection, specifically focusing on the problems surrounding wireless networks and their effects on intrusion detection. This

dissertation also highlights the problems currently associated with IDSs and proposes a model to address the current shortcomings. Next, some of the above-mentioned shortcomings have been discussed in more detail.

1.1 Motivation for this Study

As concluded, current intrusion detection systems do not adequately address intrusions in a live network environment, arguably due to the advent of gigabit-enabled network connections. Current Network IDSs are only able to effectively address a small subsection of the network traffic passing over a network (Braue, 2003). On average, most servers in use today have two Network Interface Cards preinstalled. In fact most server mother-boards come standard with two gigabit-enabled Network Interface Cards (NIC) allowing for a maximum of 2Gbps of traffic to flow in and out of the server. This amplifies the problem of current IDSs inability to cope with the sheer volume of traffic especially the packets-per-second (pps) rate. Often the packet filtering and interruptions to the flow of network packets can cause such a load as to send the system into thrashing (Dreger et al, 2004); **this is especially true if all the hosts on the network transfer a few terabytes of data each day.**

One of the most serious problems associated with current IDSs is the current lack of control in wireless segments of the network (Foong Heng et al., 2003). More and more organizations are making use of wireless network devices through the IEEE 802.11a, b and g protocols. Soon IEEE 802.16 d and e protocols will become the next buzzwords in wireless networks. These wireless technologies do not only provide connectivity to mobile users, but are also intrinsically insecure due to the broadcast nature of the technologies. Although much has been done to secure wireless networks, they are still very susceptible to a range of both active and passive intrusion attacks (Lim et al, 2003). One of the most popular wireless intrusion methods is called WarDriving, and consists of driving around with a wireless-enabled Notebook or PDA, with software installed, such as the freely available NetStumbler or MiniStumbler, which picks up the spillover from wireless devices in a network and then reports the devices' information to the "WarDriver" (Wardriving.com, 2002). **On wireless networks it is nearly impossible to**

confine radio waves to a specific area, as they can pervade walls and most objects; this is what wireless attackers and WarDrivers use to commit their attacks (Karanth & Tripathi, 2004).

Wireless networks currently have poor security implementations. The most commonly used method of securing a wireless connection is by implementing WEP (Wired Equivalent Privacy) (Hoskins, 2006). This standard makes use of a shared key Route Colonial 4 (RC4) stream algorithm that is used by the NIC to encrypt packets just before they are streamed onto the network and decrypted upon receipt by other network nodes sharing the WEP key (Nichols & Lekkas, 2002). WEP is usually implemented with either 40, 64 or 128-bit key strength (Gast, 2002). By sniffing enough traffic off the network, $\pm 5,000,000$ to 6,000,000 packets, an attacker can actually find the WEP encryption key within a few hours (Stubblefield et al., 2002; Blackstock & Sawadsky, 2005; Wi-Fi Alliance, 2005). **Without adequate internal sensors to seek out these kinds of attacks, many networks will fall victim to attacks originating from unknown sources and/or from unknown origin.**

Coupled with the above-mentioned problems, **within IDSs there is little correlation of attack data over a period of time. This can lead to possibly dangerous attacks slipping through, due to the lack of correlation between IDS hosts on a network.**

An intrusion detection model has previously been defined through research conducted at the Nelson Mandela Metropolitan University (NMMU). This model is known as the Next Generation Proactive Identification Model (NeGPAIM) (Botha, 2003). Although this model has the ability to proactively detect intrusion attacks and correlate attack data, it lacks the means to perform intrusion detection on wireless-based networks. Thus, this model needed to be updated in order for it to detect attacks occurring on wireless-based networks.

The following sections outline the reasons for and the methodology behind the research conducted, starting with the problem statement.

1.2 Problem Statement

The **primary problem** is that wireless networks currently have poor security implementations, and few, if any, IDSs have the capability to protect against wireless intrusion attacks (Hoskins, 2006; Anjum, Subhadrabandhu & Sarkar, 2004). Many organizations currently implement wireless networks, and the lack of wireless detection opens these organizations up for intrusion attacks over wireless networks.

The **Secondary Problems** researched during this dissertation are as follows:

- 1.2.1 Currently available IDSs offer little correlation of attack data over a period of time. This limits the number and types of attacks that could be detected.
- 1.2.2 IDSs need to address a large volume of data while attempting to detect intrusion attacks. The problem is that most IDSs cannot address the volume of data flowing over multi-gigabit networks; thus, many attacks slip through.
- 1.2.3 Most IDSs available currently detect attacks in a reactive manner; thus, the attacker is usually able to finish his/her attack before the administrator is aware that an attack has even taken place.
- 1.2.4 IDSs, in most cases, cannot detect mutations of attacks that already exist. This is primarily because their signatures are too specific and can only detect exact attacks with a specific pattern.

The next section outlines the primary and secondary objectives of this research.

1.3 Objective

The **primary objective** of this study is to investigate wireless intrusion attacks and the effects they have on IDSs currently available. The aim of this study is to identify whether they protect wireless network segments adequately.

The **secondary objectives** are as follows:

- 1.3.1 To state what can be done to minimize the occurrence of wireless-based intrusion attacks, such as Denial of Service, Man-In-The-Middle and Jamming attacks.
- 1.3.2 Utilization of the background information to propose a model that enables IDSs to proactively detect and halt wireless-based intrusion attacks by making use of smart agents. These agents are located at strategic points on the network, to provide choke points through which all wireless traffic must flow. The model referred to is an updated model of previous research.
- 1.3.3 An investigation into computer crime, particularly focusing on proactively detecting wireless intrusion attacks.
- 1.3.4 Creation of a prototype to determine the feasibility and effectiveness of the updates added to the NeGPAIM model.

1.4 Methodology

The methodology that was utilized for this research project comprised the following:

- The research methodology, on which this dissertation is based, is qualitative, specifically using the **phenomenological** research philosophy (Phenomenologycenter.com, 1997).
- A **literature study** has been undertaken to establish the current state of IS in a modern IT environment. The literature study has also been engaged in analysing and arguing the facts, while studying various real intrusion cases in order to identify the key aspects included in monitoring and detecting network and wireless intrusions.
- This was followed by an **investigation** conducted in order to evaluate currently available commercial packages that might combat wired and wireless-network intrusion attacks proactively.

- Next, an updated **model has been proposed, which argues** for the provision of a theoretical solution to the problem of monitoring and detection of wired and wireless-network intrusions.
- Finally, the updated **model has been implemented practically**, in the form of a basic prototype, to prove that most of the new key functionalities that the model proposes are indeed feasible in a practical sense.

1.5 Layout of the Dissertation

The dissertation consists of 8 chapters, the layout of which is depicted in figure 1.1.

Roadmap for this dissertation

Chapter 1 presents the research subject and gives background information to define the problem area. **Chapter 2** investigates the importance of computer and network security and the use of IDSs in organizations. This chapter highlights the fact that although intrusion prevention techniques are good to have, they alone are not adequate to protect the systems; thus, showing that intrusion detection systems are indeed needed. **Chapter 3** continues with intrusion detection and the problems currently associated with IDSs, both within the wired and wireless environments. **Chapter 4** provides more background on intrusion detection in wireless networks and, in particular, shows that wireless networks are becoming a big problem within organizations. This chapter also highlights that corporate network and current systems do not adequately cater for wireless intrusion attacks. The chapter concludes with a short discussion on wireless intrusion detection, and what could be done to improve the problems currently existing between wireless networks and Intrusion Detection Systems. **Chapter 5** proposes an updated model and focuses on how proactive actions can minimize the effects of wireless-based attacks. **Chapter 6** forms the heart of the dissertation and discusses the previously proposed model in more depth, as to what should be changed in it so that the model can also be implemented in a wireless environment. Finally, the chapter concludes with a discussion of how the changes have been implemented. **Chapter 7** provides results of case studies

and practical experiments conducted to support the proposed updated model. **Chapter 8** concludes the dissertation by summarizing the key aspects and what was achieved by the research project. Finally, a short discussion is provided on further research possibilities.

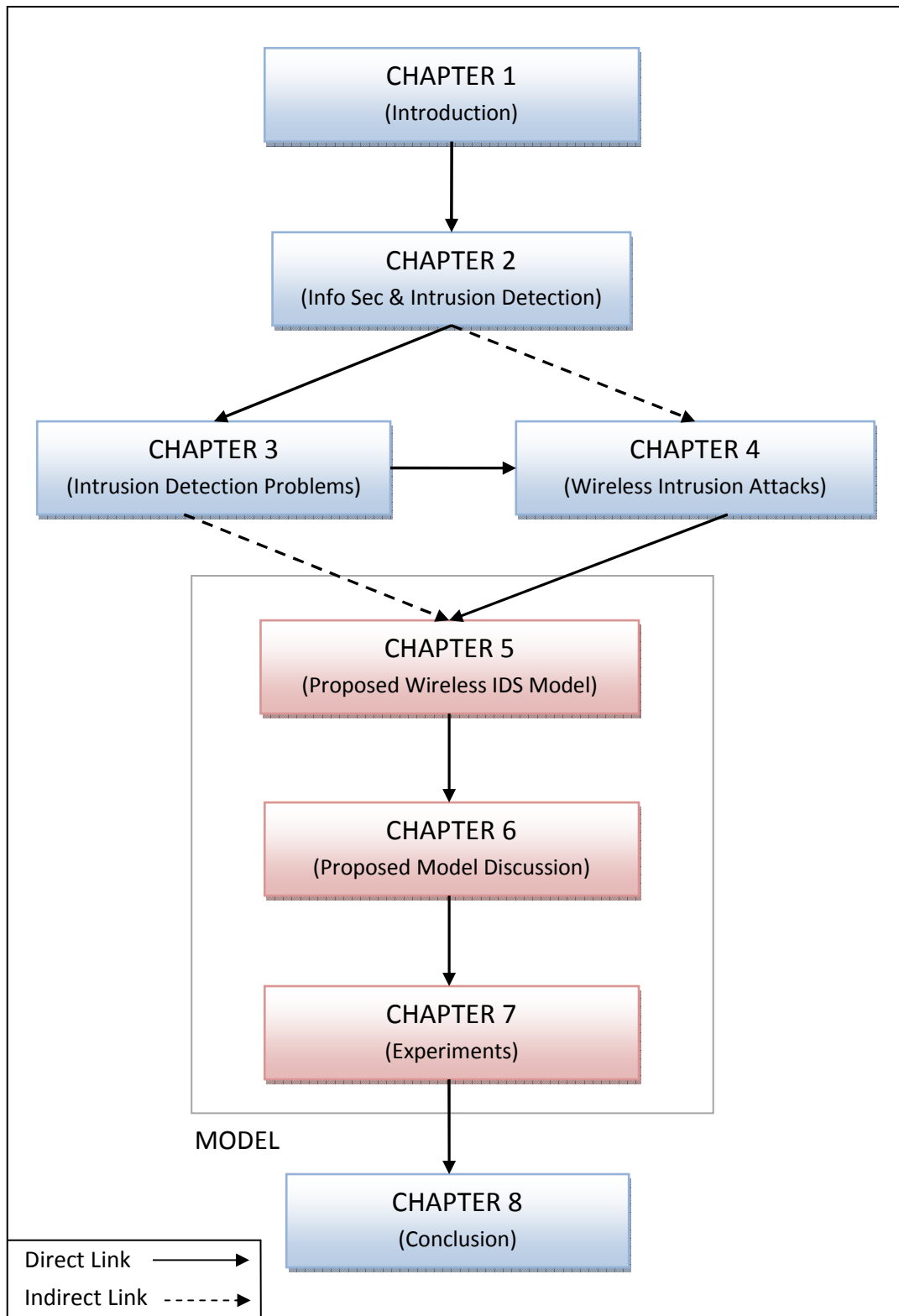


Figure 1.1: Proposed layout of dissertation

1.6 Conclusion

The main objective of this chapter is firstly to introduce the problem associated with information and computer security, and secondly, to provide a roadmap to follow the rest of the dissertation. It is quite clear from the introduction that there are a few problems currently associated with IDSs. The most important problem identified is that IDSs do not adequately cater for wireless-based attacks. Detection of wireless-based intrusions is an important characteristic, and for any IDS to be lacking in this area could spell disaster. Therefore, the main objective of this dissertation is to search for new means and ways to improve existing IDSs, or to define a new methodology, which can ultimately be used to implement an IDS that performs both wired and wireless intrusion attacks detection.

The next chapter is an extension from this chapter and discusses Intrusion Detection (ID) in general. Included in Chapter 2 is an introduction to the core of this dissertation, namely, ID. The background information gained in this chapter is ultimately used in the definition of a model, which can do proactive and dynamic wired and wireless intrusion attacks detection.

Chapter 2

Intrusion Detection in General

2.1 Introduction

One of the best-known statements in the world of information and computer security is, “Access to information is power” (Nagaraj, 1999). In today’s world, information and access to information are becoming more critical than ever, with an ever increasing demand for access to information.

The proof of the last mentioned statement can be seen in the everyday use of e-mail, the daily use of SMS and MMS for instant messaging, the rapid advances in communication technologies, and the speed at which current networks run. Unfortunately, along with these tools/technologies, problems, there is also a spread of worms and viruses. This is especially true when considering that most corporate networks have a backbone running at 1Gbit, and all workstations usually running at 100Mbit, which allow worms to spread to every machine on a LAN in a matter of seconds. Add to this the fact that most large corporate organizations have multi-megabit links to the Internet, and the problem is further compounded.

The volume and speed at which malicious hackers and cyber-terrorists are releasing malware, viruses and attacks aimed at crippling organizational information systems continue to escalate. From the first Internet worm (known as the Morris worm and named after its creator Robert Morris back in 1988), to today’s worms, such as Sober, Slammer and Nimda, the speed at which worms and viruses affect computer systems has grown exponentially (White & DiCenco, 2005). In every way, the next 20 years will bring more of everything: more threats, more attacks, more resources-at-risk, more interconnection, more communication and more emergencies. This is a simple projection from the growth trends of the past 20 years (Longstaff, 2004). Thus, one can

see the need for software and hardware that will proactively and quickly detect the presence of worms, viruses and other malware as they enter the network, as well as a range of hacks.

Seeing that computer security has become so important in today's business environment, the first part of the chapter discusses the computer security concepts with the main aim of providing one with a general background of computer security. Next, real-life computer security problems will be looked at, followed by a discussion of mechanisms allowing protection from these security problems. Finally, the chapter concludes with an introduction to ID and IDSs including the history behind this technology.

2.2 Computer Security Concepts in General

Computer security has evolved over the past 30 years that computers have been connected via networks. In the olden days, a mainframe would have been locked in a room, and it would be considered secure from anyone wanting to cause damage to it or the data stored therein. These days, computers, and the information they hold, are only as secure as the latest security patches, which as soon as they are released, mean there are almost certainly already mutations of them on the Internet. So, security officers in organizations are fighting a battle where they are always one step behind the attackers.

Computer security can be thought of in terms of four pillars on which the security of an organization's computer systems and information should be based: confidentiality, integrity, availability and accountability (Goguen et al., 2002).

- **Confidentiality:** Is the organizational control of who gets access rights to information on a corporate network and computer systems, and can be defined as the quality or state of preventing disclosure to unauthorized individuals or systems (Whitman & Mattford, 2003).
- **Integrity:** Can be defined as the quality or state of information being uncorrupted, whole, and in its original undisrupted state. The integrity of files and data can be

authenticated via the use of hashing algorithms used on the files or data to ensure their wholeness.

- **Availability:** Is defined as the ability of users to access critical information and to do so in an unobstructed and timely manner. However, information availability will not be granted to all users, only those with sufficient rights (Whitman & Mattford, 2003; Goguen et al., 2002).
- **Accountability:** Can be defined as knowing who has had access to information or resources on the corporate LAN (Lampson, 2004).

Computer and information security have been around for quite some time and have very important roles to play in organizations. Computer security most commonly refers to the controls and measures, e.g., firewalls, antivirus and access controls, etc., put in place to protect computer systems and the information stored within. Information security, on the other hand, is most commonly thought of as the policy and governance of the uses and rights to information stored on an organizational network or computer system. Both information and computer security are extremely important and go hand in hand. In fact, most of the time, they are perceived as being identical, because they are so closely linked.

Information security, when implemented properly, performs a couple of very important functions for an organization. Firstly, it protects an organization's ability to function. It also safeguards the data that the organization has collected over time, as well as its technology assets, against malicious damage. In information security, there are three elements that allow one to determine if an attack is possible, and then, offering protection from such an attack. These elements are assets, vulnerabilities and threats (Ciampa, 2005; Goguen et al., 2002).

- **Assets:** These are organizational elements that are to be protected from damage. They can be either a logical element, such as data or information, or a physical element, such as an employee or a computer system.
- **Vulnerabilities:** These are flaws or weaknesses in the system that allow unauthorized access to the organizational assets and range from bugs in software, to doors with broken locks, windows left open, etc.

- **Threats:** These are categories, objects or people that pose a potential danger to an organizational asset, like computer viruses, or physical events, like the theft of a server (Goodman, 2003; Goguen et al., 2002).

Goguen et al. (2002) state that risk assessment is the first part of any risk management methodology, and the mitigation of risk to an organization's information assets is one of the primary reasons any organization implements information security policies, so that they will have plans in place in the event that a risk becomes a reality. No matter what business an organization is in, it is impossible to not collect and store information on customers, patients, students or even employees, and thus, there will always be risk involved from identity theft, as well as others, like risk of corporate espionage, fire or even human error for the organization as well (Federal Trade Commission, 2004). Risk can be defined as the possibility that a threat or multiple threats will exploit a vulnerability existing in a system and these impact on an organization's asset or assets, causing loss of information, assets or revenue directly (Whitman & Mattford, 2003; Hash, 2002).

Below are the nine steps that should be followed during a risk assessment. According to Goguen et al. (2002), one should pay specific attention to Steps 4 and 8 when considering risk analysis; therefore, the next section focuses mainly on these steps.

1. System Characterization
2. Threat Identification
3. Vulnerability Identification
4. Control Analysis
5. Likelihood Determination
6. Impact Analysis
7. Risk Determination
8. Control Recommendations
9. Results Documentation (Goguen et al., 2002).

Steps 4 to 8, as noted previously, are very important in the determination and control of risk analysis and control. Control analysis (**Step 4**) is the process whereby management carefully checks over the controls implemented, or controls that are planned for implementation. This is

to minimize the probability of a vulnerability being exploited. After careful analysis, the organization will have an overall likelihood rating, indicating the probability of exploits in the vulnerability zone. Likelihood determination (*Step 5*) has to do with the categorization of a specific threat exploiting a specific vulnerability and can be categorized as having a High, Medium and Low probability rating. Impact analysis (*Step 6*) allows an organization to realize what would happen if any specific vulnerability was actually exploited. These negative outcomes can be categorized in terms of losses of confidentiality, integrity and availability to the system. The system must now be adapted by running a risk determination (*Step 7*). This step allows an administrator to assess the overall risk to the IT systems in an organization and can be expressed in terms of magnitude and likelihood of attacks and adequacy of controls in place. *Step 8* is an important next step as it considers all the risks and puts in place the controls that could mitigate or illuminate them. (Goguen et al., 2002).

Information security controls are needed and implemented to mitigate the risk an organization faces, although risk can never be eliminated in a real-world computer system. This can be seen by looking at the number of lines of code in Microsoft Windows 2000, which has an estimated 30 million lines of code. The estimates on bugs in programming code lie between 5 and 15 bugs per 10,000 lines of code. Thus, one can see that with the Windows 2000 code, there should theoretically be around 150,000 defects; most of which will probably never be found by programmers (Lynn, 2002).

Below are some information security control categories. The controls are divided into three general classes, and each class is divided further into generic families. Each family has many different security controls allocated to it, as these controls will fall under the specific family's umbrella. The three general classes that are available are operational, management and technical controls (Katzke et al., 2005; Goguen et al., 2002).

Class	Family
Management	Risk Assessment
Management	Planning
Management	System and Services Acquisition
Management	Certification, Accreditation and Security Assessments
Operational	Personnel Security
Operational	Physical and Environmental Protection
Operational	Contingency Planning
Operational	Configuration Management
Operational	Maintenance
Operational	System and Information Integrity
Operational	Media Protection
Operational	Incident Response
Operational	Awareness and Training
Technical	Identification and Authorization
Technical	Access Control
Technical	Audit and Accountability
Technical	Systems and Communications Protection

Table 2.1: Security Control Classes and Families (Katzke et al., 2005; Goguen et al., 2002)

Goguen et al. (2002) say that there are two specific categories of controls: a control can be either **preventative** or **detective**:

- Preventative controls inhibit attempts to violate or circumvent security policy, and include controls such as encryption and authentication.
- Detective controls warn of any attempts on or violation of security policy on a system. Some controls include: intrusion detection, access logs and checksums.

While the many preventative and detective controls available to organizations protect against and warn of attack, information security backup controls need to be installed and run alongside the regular security controls, keeping important data safe in the case of disastrous events, or in the event that an attack actually breaches regular security controls. IS backup controls should include testing of all backup files, storing offsite backups of all mission critical information, backup storage in fire resistant housings and selectively using system backup restoration as part of testing organizational contingency plans (Katzke et al., 2005). Setup and utilization of security controls are very important parts of the information security process and, when done correctly, can aid in the continuity of business processes in the event of fires, network attacks and many other environmental and human error crises.

Even though security and backup controls may be in place, there is still some degree of risk involved, which should be brought to levels as low as possible. Risk mitigation is one of the most important parts in the risk-management process and needs to be done properly in the early stages of risk management, literally, as soon as all the risks have been identified. Risk mitigation needs to incorporate prioritization, evaluation and implementation of risk-reducing controls as identified before. Risk-mitigation controls need to take into account cost, most appropriate controls and controls that would create minimal adverse impact to the organization. All of these need to be researched so that the best balance may be found. There are six risk-mitigation options available to senior management during the process of protecting their organization's assets (Hash, 2002):

- **Risk Assumption:** Acceptance of a possible risk to an asset and to continue use of the system as normal, or the implementation of controls to bring the risk down to acceptable levels.

- **Risk Avoidance:** Avoidance of risk by total elimination of the risk cause, or by shutting down certain functions of the system which cause the risk.
- **Risk Limitation:** Limitation of the risk by use of controls to limit the effects the threat agent has on the system.
- **Risk Planning:** Management of risk by development of a risk-management plan that utilizes, prioritizes and maintains controls.
- **Risk Transference:** Transferring the risk by utilization of options that may compensate for the loss, e.g., insurance schemes.
- **Risk Research and Acknowledgement:** To acknowledge that there are flaws in the system, by researching controls to address the vulnerabilities; ultimately lessening the risk (Goguen et al., 2002; Hash, 2002).

As stated before, it is nearly impossible to fully eliminate risk without actually rendering the system that contains the risk unusable (Hash, 2002). Because of this, one has residual risk, which is the risk left over after all attempts have been made to mitigate the risk associated with the specific system. Residual risk is accepted by management only after it is at a satisfactory level. If the residual risk is not at a satisfactory level, then management would have to start at the beginning of the risk-management cycle and use alternative controls and methods.

Proper incident response is another important function within the information security framework. It is all well and good having a security policy and many controls put in place to minimize the impact of an incident, but one still needs a plan in place to handle an incident further, after it has taken place. Take, for instance, the analogy of a home security system; unless one has armed response to an intrusion on one's home, one's home security system does little more than annoy one's neighbours (Patzakis, 2003). This is where incident response comes in, and one of the best ways to create an incident response plan is to contact one's local Computer Security Incident Response Team (CSIRT). CSIRTs are organizations that release technical documents and provide help to anyone who is currently recovering from an incident, or is setting up an incident response system. CSIRTs are available to anyone needing their assistance via e-mail or telephone, CERT/CC was one of the first organizations providing this kind of service (West-Brown et al., 2003).

ISO17799 recommends that an enterprise or organization establish procedures to ensure a quick, effective and orderly response to security incidents that may occur. These procedures must include the following (Guidance Software, 2003):

1. Identification and analysis of the cause of the incident.
2. Planning and/or remedies that will prevent the reoccurrence of the incident.
3. Collection of audit trails and similar evidence.
4. Communication with those affected by or involved in recovery from the incident.
5. Reporting the activity to an appropriate authority.

Thinking about the recommended procedures leads one to the conclusion that there are three reasons why the affected organization should collect data about the attack that may have occurred. These are as follows (Guidance Software, 2003):

1. Internal problem analysis.
2. Use of evidence in relation to potential breach of contract, breach of regulatory requirement or in the event of civil or criminal proceedings.
3. Negotiation for compensation from software and service suppliers.

Now that one has the background on risk, risk management and various types of security controls available to organizational management, the next step is to determine where these can be implemented. Thus, the next section introduces real-world computer security problems organizations are currently faced with.

2.3 Real-World Computer Security Problems

As society's need to access critical information increases, and this includes access to banking, medical and personal information, so does the need to secure the systems, software and the information that is to be accessed by legitimate users to ensure that illegitimate access is eliminated (Rogers, 2004). As discussed in the previous section, this is done by implementing information security and policies that prevent unauthorized access. The problem is that until a hack, virus, Trojan horse, etc., is actually found for the first time, there is not much that can be done to protect one's organization from it.

New attacks come out virtually on the hour, and most organizations cannot be 100% sure that they, in fact, have the latest updates, patches and service packs installed. This is partly because vendors do not always alert one to the latest updates (Rogers, 2001). These attacks are not limited to any one industry, and, in fact, occur in sectors from medical, governmental and retail, all the way through the spectrum to academic institutions. Coupled with new compliance laws set out by government, it becomes increasingly important to adequately protect organizational information (Radvanovsky, 2004). As can be seen, no one is safe from malicious individuals in their quest to cause damage to valuable information assets.

As previously stated, no organization with a connection to the Internet is completely safe from intrusions and attacks, and below are listed a few of the types of attacks and potential problems that organizations may have to protect themselves against (Hansman, 2003).

- **Mutant Attacks:** One of the problems currently associated with computer security is that attacks can and are mutated to get past virus scanners and other defence mechanisms. Take for instance, the W32.Sober worm. Symantec has a removal tool that can remove 17 variants of this worm, all mutants of the original W32.Sober worm, first discovered on October 29th 2003 (Symantec, 2006b).
- **Script Kiddies:** Another problem is that most seasoned hackers publish how they completed an attack on hacker websites, such as www.rootshell.com. Some even release tools that make it easy to take advantage of specific bugs in a system; thus allowing unskilled people to hack a server or website as if they were an elite hacker. These “script kiddies”, as they are known, can cause damage that before could only be done by highly skilled individuals.
- **Eavesdropping:** Eavesdropping can be a problem at many levels, allowing the theft of information, e.g., local, at transmission, wireless spillover, tempest hardware and hardware errors. All of these allow hackers to gain access to data that would normally not be accessible to them.
- **Malicious Software:** Malicious software, such as viruses, Trojan horses and commercial software used for inappropriate activity.

- **Inadequate Management:** Many networks suffer security problems as a direct result of administrators and network managers not knowing their systems properly, or not doing system updates and critical security patches in a timely manner.
- **Ignorant users:** Ignorant users, be they end users or super users, and the lack of security training and knowledge can put an organization's network security at serious risk of malicious users taking advantage of the system.
- **User Error:** Mis-configuration and other user errors can lead to inadvertent holes in network security, opening a network up to attacks that were previously patched and secure.

These are some of the more prevalent types of security problems, and many do have remedies; however, many only require the education of users allowing them to better understand the importance of security to the organization.

2.4 Protection Mechanisms

When studying real-world computer security problems, it became clear that there is a need to protect the valuable software, networks and information. For this reason, there are many protection mechanisms, all implementing highly sophisticated methodologies. In this section the various protection mechanisms have been discussed, as well as how they are implemented to protect an organization from attacks, be they internal or external, on the network. Protection mechanisms allow for the protection and preservation of information confidentiality, integrity and availability by controlling user access to the information asset (Gonzales, 2005). There are many different categories of protection mechanism and below are listed the main categories (Katzke et al., 2005):

- Access control
- Awareness & training
- Audit & accountability
- Certification, accreditation & security assessments
- Configuration management

- Contingency planning
- Identification & authentication
- Incident response
- Maintenance
- Media protection
- Physical & environmental protection
- Personnel security
- Risk assessment
- System & information integrity
- Systems & communications protection

Protection mechanisms, such as physical and environmental protection, are aimed at securing offices and computer centres by the use of locked doors, and include physical access control mechanisms like biometric lock controls, or limiting access to only those granted access by an administrator. On the other hand, copy protection, encipherment, digital signatures, data integrity mechanisms and the likes are all forms of media, systems, information and communications protection mechanisms, allowing the authenticity and confidentiality of data to remain intact and when data is no longer needed, securely destroying it. There is yet another mechanism type: that of protection via software and applications, such as firewalls, IDSs and routing controls to provide safety of information whilst it is in transit over the network, including thwarting attempts of external hackers and internal users with insufficient rights gaining access to confidential information. Many of these software-driven mechanisms are built with decision-making abilities for attack, error, user and access rights determination.

A hard lesson that has been learned by many systems administrators is that of assuming that with the wide range of security mechanisms implemented in their networks, an intruder could not possibly break through them all. This way of thinking is foolhardy as all it takes is one mistake on a configuration, and the attacker has full access. For this reason, an IDS is a much-needed protection mechanism in the detection of intrusions that occur, even though there are many protection mechanisms in place (Bace, 2000).

Currently, technical controls, such as IDSs, work in isolation to access control mechanisms, as discussed in *Section 2.5* and *2.6*. From this, one can see that intrusion detection is passive; therefore, it only alerts an administrator in the event of an intrusion or anomalous event. With all this said, IDSs are still needed for access control to work in an organization. With the IDS alerting the administrator to intrusions, he/she can patch and update his/her access control scheme (Alpcan & Bascar, 2004). Research is currently being done in the field of Intrusion Prevention (IP), explained in detail later in the chapter.

2.5 Intrusion and Intrusion Detection Concepts

As discussed in the previous section, there are many protection mechanisms, many of which are software based, including that of IDSs. This section introduces intrusion and intrusion-detection concepts that recently have been tending towards IPSs (Intrusion Prevention Systems) over the past few years, as the need to catch intruders quickly or in the act increases. This research currently is the way intrusion detection research is heading: to proactively identify and possibly stop an attack. Hopefully, these technologies will save organizations time and money when hunting down hackers.

Computer systems / networks have been around for a few decades now and, as with any technology that houses sensitive information, certain people would like to have this information. As long as computers have been around, so have the people devising ways to break into computer systems and networks. These unauthorized access attempts are known as intrusions and can be defined as “a set of actions aimed at compromising the security goals, namely confidentiality, integrity and availability of a computing / networking resource”. The detection of intrusions (Intrusion Detection) can be defined as the process of identifying and responding to intrusion activities (Petrovic, 2005).

Modern intrusion detection and intrusion detection systems’ research are moving towards a new type of methodology: that of proactive intrusion identification, allowing intrusions to be identified before any damage is done, thus catching intruders in the act and preventing any future attacks of this kind from happening again. This new form of intrusion detection is known as

intrusion prevention, as it not only detects attacks, but prevents many from occurring at all (Doctor, 2004). The main difference in methodologies is that IDSs are passive security solutions as opposed to IPSs, which are active security solutions (Beal, 2005). IPSs are currently considered the 'next generation' of IDSs.

Another newly added intrusion detection and prevention arena is that of wireless IDS and IPS systems. The reason for this research is that there are many attacks that only target wireless-enabled network segments and not their wired counterparts. These attacks include creating of 'rogue' access points, war-driving and flooding APs with bogus association requests (Petrovic, 2005). Wireless IDSs/IPSs have evolved over the past few years from running primarily on hosts to systems that have remote agents and network sensors, located in strategic network locations, allowing administrators to detect, stop and even prevent attacks on their wireless LANs.

There are currently two methodologies to which IDSs ascribe, the first is that of misuse detection and the second is anomaly detection, the next two subsections give an overview of these methodologies.

2.5.1 Misuse detection in general

As previously mentioned, misuse detection is a common method used in detection of intrusions/attacks in IDSs and is also used by antivirus software to detect viruses. In misuse detection, the IDS collects the data that needs to be scanned for an attack, and each piece of data is compared against an attack database kept by the IDS, containing signatures of all attacks that may have already been performed.

These attack signatures need to be updated either manually by the administrator by utilizing the IDS vendor's website and downloading the new signatures or alternatively, the signatures can be downloaded and installed automatically by the IDS script. Misuse detection has the important advantage of allowing for a very low false alarm or false positive rates, as any attack that it detects has been compared to known intrusive behaviour and, therefore, should be flagged as an intrusion attack (Carter, 2002). Another benefit of misuse detection is that it is already, directly after installation, able to detect attacks out the box, via the signature

database. Other technologies, such as anomaly detection, require the system to be trained before it is able to detect attacks (Carter, 2002).

The main disadvantage with a misuse detection engine is that the systems and networks it protects are only as secure as the latest attack definitions. Intruders and hackers are aware of the last mentioned disadvantage of misuse detection; therefore it is important to research for additional solutions and/or additional approaches.

The next section introduces anomaly detection as the second methodology currently utilized in IDSs.

2.5.2 Anomaly Detection in General

Anomaly detection in intrusion detection systems is behaviour-based and compares a profile of allowed user behaviour on a system to the actual live user behaviour, with any deviation from the profile being flagged as a potential attack (Ning et al., 2005; Maselli et al., 2002; Bace, 2000). Anomaly detection works on the idea that if something is out of the ordinary, it is more than likely a potential threat to the system. Systems, based on this paradigm, have to be trained in order to recognize which behaviour is normal and which is not and thus unacceptable (Maselli et al., 2002).

One of the main advantages of an IDS based on anomaly detection is the fact that it can detect attacks that are not yet known or do not have signatures available for the attack. In the case of many internet worms, this is a good thing to have as they mutate through many versions to defeat IDS signatures. The downside to anomaly detection, which must be taken into consideration when defining any detection methodology, is that there are usually many false positives. This means the system detecting user behaviour as an attack, when in fact the user has just changed his work habits slightly (Valeur, 2004; Maselli et al., 2002).

The main criticism with anomaly detection engines, however, is the fact that an intrusion can fly under the radar. This is done by the user/intruder performing his attack step by step, slowly over a long period of time. By doing so, the anomaly detection engine will be trained to accept this behaviour as normal, and it will not be taken as an intrusion. This is able to

take place because anomaly detection engines need regular training to allow the engine to function better as users and other variables on the system are updated (Tan et al., 2002).

This weakness needs to be addressed by a protection mechanism that prevents attackers from exploiting the vulnerability. This protection mechanism existed in the previous NeGPAIM model and is still very applicable to the updated NeGPAIM-W Model. The mechanism is referred to as the safe mechanism in this dissertation. This safe mechanism is implemented in the fuzzy engine and will be explained in Section 6.3.1.

In the next section, the history of intrusion detection, and IDSs in particular, have been discussed in detail.

2.6 History of Intrusion Detection Systems

As can be seen from the previous sections, intrusion detection systems have come a long way since their beginnings in the 1970s when the American Department of Defense (DOD) and the American government, began to see the proliferation of computer usage within their ranks. This proliferation of computer usage began to scare the upper echelons of the military and government. Since auditing of computer systems had already begun, and the audit community had much experience in the tracking and logging of activities taking place on computers, they were enlisted for their knowledge, thus assisting the military to track computer usage. Between the years 1977 and 1978, meetings were set up by the National Bureau of Standards with Electronic Data Processing (EDP) auditing organizations, both governmental and commercial (Bace, 2000; Bruneau, 2001).

James P. Anderson was the first to realize the need for an automatic audit trail review to support the goals of computer security. He published a study for the US Air Force in 1980 that is considered the birth of Intrusion Detection and conceived the notion that misuse could be “detected” as well as other user specific activities. His paper was based on a client that had a stringent security policy, and all audit logs were checked manually by security staff, Anderson

proposed a taxonomy for classifying risks and threats to computer systems that differentiated problems by source, either internal or external (Bace, 2000; Innella, 2001).

Dorothy Denning and Peter Neumann researched and developed a model for real-time intrusion detection (between 1984 and 1986). The model proposed was an expert system, named IDES (Intrusion Detection Expert System) (Denning, 1986). Their research was funded by the US Navy's Space and Naval Warfare Systems Command (SPAWARS) and proposed a correlation between anomalous behaviour and misuse in systems. Denning went on to write another seminal research paper on IDES and intrusion detection in 1987. Denning and Neumann's 1986 model was implemented in a prototype by SRI International and was completed in 1992. During the 1980s, many prototypes and models of intrusion detection systems were written, primarily because of the works of Anderson and Denning (Bace, 2000; Innella, 2001).

The next great advancement in intrusion detection systems came in the form of hybrid systems, integrating both network-based and host-based intrusion detection systems into a more holistic IDS. This took place in the early 1990s and continues today, with new advancements in technology tending towards early detection and prevention of intrusions with Intrusion Prevention Systems (Bace, 2000; Bruneau, 2001).

2.7 Conclusion

As seen through the research of James Anderson, Dorothy Denning and other pioneers of intrusion detection, intrusion detection systems are indeed not only needed, but are, in fact, an indispensable part of any organization's arsenal of tools to detect and stop intruders, whether internal or external, to the organization. As intrusion detection evolves into wireless networks and into intrusion prevention, administrators will gain more control over their systems and networks, allowing for the information therein to remain safe from those wishing to steal, corrupt or even destroy an organization's most valuable asset: its information.

The next chapter take an in-depth look at the ID and IDS world by showing a typical architecture of IDSs as well as demonstrating the protocols on which networks are dependent. This will not

only provide knowledge about how the networks operate, but also indicate the need for security mechanisms, like intrusion detection systems, in a Defence-in-Depth strategy. Defence-in-Depth can be defined as *“Having multiple layers of defense much like the layers of an onion. Each layer complementing the others by providing a different type of security mechanism but achieving the same result (defenceindepth.com, 2007).”*

Chapter 3

Important Network Standards and Attacks

3.1 Introduction

As indicated previously, intrusions and malicious users have always been a part of computing security history. Therefore, the protection of information resources should be one of the primary objectives when developing an organization's information system. One way to protect information resources is through the use of tools that are available to organizations to boost their security and mitigate the risk of attack to acceptable levels. One of these is an Intrusion Detection System (IDS). This chapter focuses mainly on networking technologies and standards.

To understand how to solve a problem, one first needs to fully understand the environment in which the problem exists, as well as the problem itself. For this reason, it is necessary to understand the various network technologies on which a typical IDS resides. This allows one insight into possible problem areas within each of the various technologies, as no technology is flawless. With this information, it is possible to determine what the architecture of an IDS should contain and how best to design the IDS.

When thinking about the architecture of any tool, one first needs to know the environment in which the tool will operate, and IDSs are no different. The first part of this chapter explains where networking started, where it currently is, and where it is heading. The chapter moves on to discuss the OSI reference model and how its layers are implemented. The physical networking standards are also looked at, including new standards, such as WiMAX and others. All the aforementioned groundwork leads to the crux of the chapter: a discussion about hacks and attacks based on wired and wireless networks.

3.2 Networking in General

Computer networks have been around since the late 1960s and have taken over the way organizations do business, as well as making it possible for any person to do business with anyone, anywhere in the world. This is done in an instant; the process of doing business becoming easier and receiving payments faster than ever before (Lesonsky, 2006). Computer networks inherited their basic functionality from the pre-existing telephone networks. Computer networks are a logical evolution from telephone networks, seeing as how both telephones and computers are extremely important in the way modern business is done. Networking can be defined as “the interconnection of workstations, peripherals (such as printers, hard disks and scanners) and other devices” (Amato, 2000).

This information is needed as a background to further chapters of this dissertation, allowing insight into the direction in which computer networks are moving, including knowledge on vulnerabilities and attacks against networks. In this section, the history and future of computer networks is discussed.

3.2.1 History of Networking

High-speed data communications were developed along with the systems supporting them in the 1960s, when some of the first modems were frequency shift keying modems. These modems were only capable of transmitting a data stream of 1,200 bits/s over dedicated wires, or over the public switched telephone network at even lower speeds of 300 bits/s (Freer, 1988). Over the next ten years, data-transmission speeds over the public telephone network increased to average speeds of 2,400 bits per second or less. This was used to send data to a remote host for processing by an application program, so these were host-centric networks. To send data to another city or town, the data would first have to be sent to a remote host and only then would it be relayed to the destination host, a process known as terminal-to-terminal communication. This communication ran via proprietary protocols, making it difficult to communicate between organizations running different protocols (Wilder, 1998).

The next big leap in internetworking computers was the conception and wide-scale adoption of Local Area Networks (LANs), allowing for the interconnection of computers,

workstations and terminals in a building or group of buildings. This emerged in the 1980s and began a trend in networking towards high-capacity low-cost networks. This trend is still applicable in modern LAN environments, allowing truly distributed computing to become a reality, as well as various other technologies people today take for granted, such as e-mail, online shopping and online banking (Freer, 1988).

Today, according to Stallings (1997), the computer-communication networks revolution has reached a new level and can best be described as follows:

- There is no fundamental difference between data processing and data communications equipment; the lines between equipment are becoming blurred.
- There is no fundamental difference between data, voice and video communications.
- The differences between single-, dual- and multi-processor computers, LAN, WAN, MAN and other long-haul networks are becoming indistinct with each passing year.

Next, modern networks and the technologies that drive them are discussed.

3.2.2 Modern Networks

Computer communications networks have progressed rapidly since their inception in the early 1960s. Today, one has faster speeds, greater connectivity and even wireless and fiber optic communication technologies, all allowing for the better communication of data from one computer to the next (Coffman & Odlyzko, 2001). Networks today are an indispensable part of the way most organizations conduct business, whether it be in their online presence, such as in banking, ordering materials, order processing or any other business process, networks enable them to perform these processes more efficiently (Lesonsky, 2006). Most modern networks are implemented using non-proprietary network standards and protocols, such as Ethernet 802.3 and 802.11, which overcome the earlier problems mentioned above regarding the difficulty of communication between users of different protocols.

The Internet is the biggest computer network in existence and has benefited much from the standardization of communication protocols and standards. The Internet is older than most people believe, starting in the early 1970s with ARPANET (Leiner et al., 1997). The Internet took off commercially around 1995, and today, the Internet's growth is doubling each year.

Arguments have been made that this may continue for the remainder of the decade (Coffman & Odlyzko, 2001). Today, one can access the Internet almost anywhere: in airports, shopping malls, on cell phones, on PDAs, etc.

Networks grow in use and speed at a rapid pace, and technologies, such as fiber-optics, are currently replacing copper-based networks to keep the speed of networks increasing past the gigabit boundary and beyond (Armosky & Hemenway, 2000; SafeNet, 2006). Currently, copper-based networks have been stretched to their maximum and are limited to one gigabit on Ethernet. On the other hand, fiber-optics are able to surpass this limit to speeds of 10 gigabit and even further, as research suggests.

Below is a list of the currently available physical EIA/TIA (Sheldon, 2001) cable types that are available to an organization when implementing its corporate network. Some are no longer in wide use, but are listed because they may still be in use in many organizations. Unless there is a good reason not to, Cat 5, 5e or 6 cables should be used.

- **EIA/TIA Cat 3:** UTP (Unshielded Twisted Pair) Cable. When running Cat3 anywhere in a network, then 100Base-T4 must be used as 100Base-TX can only operate at 100Mhz (Panduit, 2004). 10Base-T allows for up to 10Mbit/s data rate (ADC, 2003). It allows for 100 meters maximum cable run, terminated with an RJ45 connector and can only handle up to 10Mhz of bandwidth (Panduit, 2004). It is utilized mainly in two-line telephone networks.
- **Cat 4:** Unrecognized by the EIA/TIA, 16Mbit/s data rate, previously used in Token-Ring networks.
- **EIA/TIA Cat 5:** UTP Cable. Can be run as 100Base-TX if Cat5, 5e or 6 is run everywhere in one's network (Panduit, 2004), allowing for 100Mbit/s data rate and has to run at 100Mhz bandwidth (ADC, 2003). It allows for 100 meters maximum cable run, terminated with an RJ45 connector. Obsolete as surpassed by Cat5e.
- **EIA/TIA Cat 5e:** UTP or STP (Shielded Twisted Pair) Cable. Can be run as 100Base-TX or 1000Base-T if Cat5e, 6 or 7 is run everywhere in one's network (ADC, 2003). 100 meters maximum cable run and has to run at 100Mhz bandwidth

(ADC, 2003). This is currently the TIA minimum recommended cable for new network installations (Panduit, 2004).

- **EIA/TIA Cat 6:** UTP or STP Cable. Can be run as 100Base-TX or 1000Base-T if Cat5e or 6 is run everywhere in one's network. It allows for 100 meters maximum cable run terminated with an RJ45 connector and has to run at 250Mhz bandwidth (ADC, 2003). Backwards compatible with Cat3, 5 and 5e although data transmission speed may be less.
- **ISO/IEC Cat 7:** ScTP (Individually Shielded Twisted Pair) or STP Cable. Allows up to 10Gigabit Ethernet using a TERA connector not possible on other standards. It allows for 100 meters maximum cable run, Terminated with an RJ45, GG45 or TERA connector and has to run at 600Mhz bandwidth (ADC, 2003). Cable shielding around individual wire pairs, grouped in second shielding. Standard only in draft form currently.

Even though there are many different cabling standards providing varying data transmission speeds, there is still the problem of connectivity and portability of wired networks, leaving a gap in the market for wireless-based network technologies. Wireless communications, while not all that fast, allow one to move and have access to the corporate network anywhere, anytime, and data transmission speeds are increasing constantly, as is discussed in Section 3.4. Thus, wireless networks are undoubtedly the future of next generation networks.

3.2.3 The Future of Networks

As previously stated, wireless networks are the future of networking. Even with wireless networks, there will still be a need for faster access to information. With the ever-increasing need for faster access to information, and the amount of data needing to be transferred between computers and organizations, networks will have to adapt to meet the needs of organizations. Some of the requirements put on networks could be similar to the following:

- **Security Requirements:** Security in future network technologies should strive towards smarter Layer 2 security because as networks increase in capacity, there will be more strain on Layer 3-based security technologies. This strain could include higher false alarm rates in IDSs and high latency when using firewalls. Security in

future networks will more than likely have high-quality Layer 2 encryption built into the devices, unlike those currently available, which were an afterthought, such as WEP in wireless networks discussed in Section 3.4.

- **Application Requirements:** As application programs gain functionality, their usage of network resources usually grows too. Applications, such as online collaboration tools, may use a great deal of network resources, and future networks will need to note this. More and more video streaming applications, such as MSN Messenger and Skype are being utilized within organizations as a means of communication. They also utilize a large number of network resources when the whole organization is connected.
- **Cost Requirements:** Costs of new network technologies, as well as compatibility with currently existing technologies will also play a role. Replacing existing infrastructure would be too expensive to do all at once, so new technologies must be compatible with older networks. As more data needs to be sent/received, there are expectations that the cost of transmitting data will become cheaper via the new networking technologies. This is a trend that is explained by Moore's Law, which states, "The cost of technology declines by 50% every 18 months" (Coffman & Odlyzko, 2001).
- **Technical Requirements:** Technical requirements play an important role in the future of networking. Requirements, such as data transfer speeds, cabling types, cable connectors and networking equipment, will need to adapt and become smarter, faster and more robust as networks move forward. New network technologies will need to be scalable and have the ability to adapt easily to shifting bandwidth requirements, perhaps more than ever before.

When analyzing this history of networking, it is clear that standards and protocols are what make today's networks more robust and allow for greater compatibility than the proprietary network standards and protocols of the past. In order to implement a successful IDS, one should understand the most important standards and protocols. There are many different standards and protocols in the networking world: the ones discussed in the next section are the standards that have an impact on this dissertation.

3.3 OSI Reference Model

In the previous section, it was said that standards are needed to promote interoperability and to improve security. The OSI (Open Systems Interconnection) reference model is one such group of standards. This model was created by the ISO (International Standards Organization) and was released in 1984 to help network equipment vendors create interoperable equipment (Amato, 2000; Stallings, 1997). The OSI model was designed, using a widely acceptable and easily understood technique: layering. This Section on the OSI model allows one to understand concepts such as the OSI layers, which is needed in Chapters 6 and 7 for explanations of attack detection.

The ISO implemented the model using layering because each layer performs a subset of the operations that are required in communication with another computer or system. Each layer relies on its neighbouring lower layer to perform more primitive operations and to conceal the details of those operations from the next higher-level layer to which it provides service (Stallings, 1997). Secondly, the layered approach was chosen because it divides the interrelated aspects of the network operation into less complex elements, also preventing changes in one area from affecting the others (Amato, 2000). Below is a copy of the OSI reference model Figure 3.1.

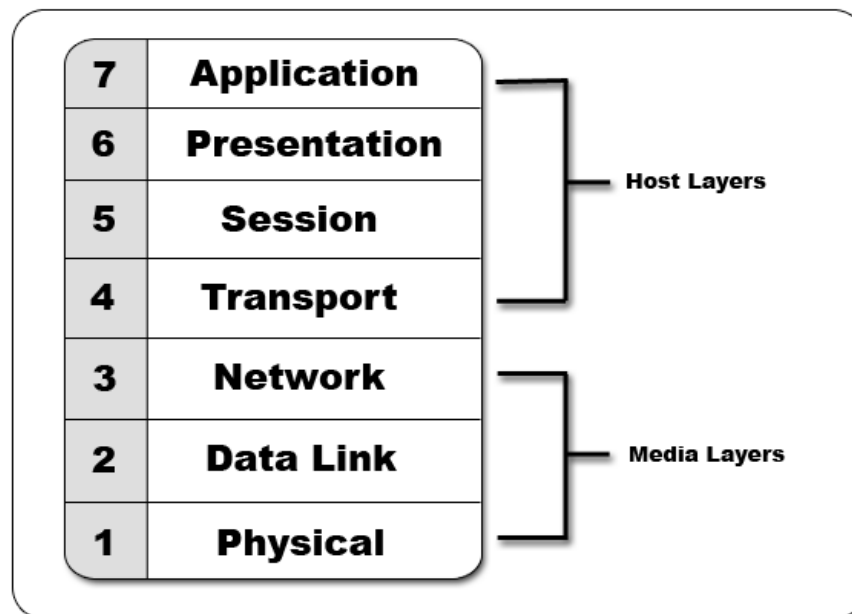


Figure 3.1: OSI Reference Model

The OSI model can be logically split up into two sections. Layers 1-3 can be grouped as media layers, as they are concerned with the physical communications' media and actual delivery of messages over the network. The second grouping, layers 4-7, is grouped as host layers as they are concerned with the accurate delivery of data to the host (Amato, 2000). Below are listed the layers and an explanation of what operations each layer is responsible for.

- 1. Physical Layer:** This layer defines the actual network media, e.g., electrical, mechanical, wires, operating, voltages and equipment specifications.
- 2. Data Link Layer:** This layer's responsibilities lie with the actual transit of data over the network medium, and it is concerned with the reliable transport of this data. It also relies on physical addressing, or MAC addresses, as opposed to logical IP addresses, used in Layer 3. Its responsibilities also include network topology, error notification and flow control of packets.
- 3. Network Layer:** The network layer provides connectivity and path selection between two end-systems. These systems may be in geographically different networks, e.g., the Internet. Routers function primarily at Layer 3.
- 4. Transport Layer:** Layer 4 is responsible for segmentation and reassembly of data into data streams. Its main focus is to relieve/shield the upper layers from transportation problems and data transmission implementation.
- 5. Session Layer:** The session layer, or Layer 5, does as its name suggests: it establishes, manages and terminates sessions between application programs. It also takes care of exception reporting of problems occurring or originating in Layers 5, 6 and 7.
- 6. Presentation Layer:** Layer 6's concern is mainly that the representation of information sent by the application layer of one system is readable by the application layer of the receiving system. The presentation layer translates between multiple data representation formats by using a common format.
- 7. Application Layer:** Layer 7 is the closest layer to the end user and provides network connectivity and services to user applications. Some of its functions are: synchronization of cooperating applications, establishment of communication partners, establishment of agreements on error recovery and control of data integrity (Amato, 2000; Stallings, 1997).

Some of the standards that utilize the OSI model can be found in the next few sections: they implement the lower layers 1 - 3 of the OSI model, whereas the upper layers 4 - 7 are implemented by protocols, such as TCP/IP.

3.4 802.3 Ethernet Standard

The previous section gave a generic background to the most common network standards. The 802.3 standard is one of the many standards created and maintained by the IEEE Computer Society. The 802.3 standard is concerned with CSMA/CD (Carrier Sense Multiple Access with Collision Detection) and physical layer access specifications. The standard covers Ethernet standards 10BASE-T, 100BASE-TX, 1000BASE-T, and they are the most commonly used Ethernet types implemented in organizational networks, running with data transmission speeds of between 10 and 1000Mbit/s (IEEE Computer Society, 2002). 10Base-T has a maximum bandwidth of 10Mbit/s, 100Base-TX, also known as fast Ethernet, allows for transmission speeds of up to 100Mbit/s and 1000Base-T (previously 1000Base-CX), also known as Gigabit Ethernet, is able to transmit and receive data at up to 1000Mbit/s (IEEE Computer Society, 2002). New additions to the standard as with 802.3ae have increased to 10 Gigabits with transmission speeds of up to 10Gb/s.

Some of the problems inherent in the 802.3 standard have to do mainly with the way the protocol advertises addresses using ARP (Address Resolution Protocol) and DHCP (Dynamic Host Configuration Protocol), which cannot be secured with a layer 3 security mechanism (Finn, 2001). The reason for this is primarily because the organization may not have legal or physical access to the layer 3 endpoint. With ARP attacks, the MAC addresses can be spoofed and data stolen off the network during transmission. This can be done using man-in-the-middle attacks (Finn, 2001). There are many other attacks that can be performed relatively easily against an 802.3 network, including VLAN (Virtual Local Area Network) hopping, MAC flooding, Spanning Tree protocol vulnerabilities and PVLAN (Private Virtual Local Area Network) vulnerabilities. Most of these attacks and vulnerabilities are performed at layer 2 of the OSI model and cannot be easily detected (Cisco Systems, 2003b).

Access control is the most common way that administrators control access to assets over the 802.3 protocol. The problem with access control is that while the latest forms of access control, such as biometrics, secure tokens, smartcards and Public Key Infrastructure (PKI), are becoming more resistant to attack (Unisys, 2005), the older forms of access control, such as username / password pairs, are still very much at risk and have many vulnerabilities associated with them, according to Gast (2004). The reason for this is that many attacks are outside of the access control mechanism's original design specification and thus, can penetrate the access control with ease (Gast, 2004). One of the ways organizations have been combating attack on older access control protocols is by the adoption of encryption tunnels, such as TLS (Transport Layer Security) and IPSEC (IP Security Protocols).

According to Gast (2004), researchers have found that there are some problems with authentication over encrypted tunnels, in that the outer encryption tunnel and inner authentication tunnel are not strongly associated. The problem is not on the part of either authentication or encryption protocols, but rather in the way they are combined, leaving them vulnerable to attack. Therefore, it can be seen that both layer 2 and layer 3 security mechanisms have their benefits, and a combination thereof would be most effective (Gast, 2004). Take for example, the combination of layer 2-based 802.1X and layer 3-based IPSec security technologies running alongside to fully protect a network running on the 802.3 and the 802.11 standards. This is highlighted in the next section.

3.5 802.11 a, b, g and n Wireless Standards

As mentioned in the previous section, the 802.3 protocol has many possible vulnerabilities; so too do the 802.11a, b, g and n standards, partly because they are based on the 802.3 standard. There are many more vulnerabilities and problems associated with running an 802.11 network, primarily because it is a wireless network, and airwaves are not as easy to protect as wired networks (Owen & Karygiannis, 2002). This is due to the dispersal of radio waves not always being controllable, so further attackers need not tap into a wire to gain access (Owen & Karygiannis, 2002). This section will describe some of the problems associated with the wireless

standards 802.11a, b, g and n, as well as some case studies and flaws within the Wired Equivalent Privacy (WEP) protocol.

Wireless LAN is one of the most common uses of wireless technologies for data transmission. Speeds vary between 1 and 54Mbit/s and, with some compression technologies, data transmission can be the same as their wired counterparts. With the arrival of the 802.11n standard sometime in 2007, the speed of wireless LANs will be boosted to 540Mbit. The frequencies that the 802.11 standards run on are as follows: 802.11a runs on the 5 GHz range, 802.11b on 2.4 GHz and 802.11g on 2.4 GHz (Owen & Karygiannis, 2002). The average range of wireless LAN equipment, including access points (APs) and wireless cards for notebooks and desktops, is around 30 to 100m, but the signal can be amplified much further (Flickenger, 2002).

Radio frequencies are regulated; therefore, an organization's wireless communications are limited to certain frequencies by law. In South Africa, communications are regulated by ICASA (Independent Communications Authority South Africa), in the United States by the FCC (Federal Communications Commission) and in Europe by ERO (European Radio-communications Office). Usually the frequency bands 5GHz and 2.4GHz are for use in wireless networks without a licence in most countries (McCullough, 2004). This, coupled with the broadcast nature of radio signals (Interlink Networks, 2002; Aruba Networks, 2004), makes it very easy for passive eavesdropping on a wireless network to take place. This is especially true seeing that the hacker already knows which frequencies the intended target will most likely be running on, thereby making his/her attack easier, and also because of lower quality encryption, usually 56bit keys instead of 128bit keys.

Wireless networks are costly as an initial investment, but their flexibility make them, more often than not, cheaper to run than wired Ethernet (McCullough, 2004). Often this saving of money by the organization may make the vulnerabilities and potential threats associated with wireless technologies worth the risk. The increased mobility of users may also help justify the additional risk, as users can connect to the network from virtually any location in the organization that has wireless coverage and will not have to find a network port to get connected. It will be difficult for wireless networks to replace wired Ethernet networks, because the speeds at which data can

be transmitted over wireless is still nowhere near the speeds of its wired counterpart (Gast, 2002).

A new initiative from a group of companies to reduce the gap in data speed is the definition of the 802.16 wireless standard. This standard is also known as WiMAX and will provide roaming data networks, which, like those of cellular telephones, will be indispensable to organizations in their day-to-day business networking. WiMAX is discussed in detail in the next section.

3.6 802.16 WiMAX Wireless Standard

As highlighted in the previous section, the new buzzword in wireless networking is WiMAX. Worldwide Interoperability for Microwave Access (WiMAX), also known as 802.16, is a fairly new technology that will allow true broadband Internet over a wireless medium. It could also be used to replace any existing wireless networks that are deemed too slow. WiMAX is a point-to-multipoint connection, according to Vaughan-Nichols (2004). Point-to-multipoint microwave connections have been in use for years by companies, such as Alcatel and Siemens in proprietary forms. With the WiMAX standard high-bandwidth, wireless point-to-multipoint connections will be standardized allowing any WiMAX-based equipment to connect and access any access-point, no matter the brand (Vaughan-Nichols, 2004).

According to Vaughan-Nichols (2004), WiMAX will more than likely provide a backbone between buildings and areas for 802.11-based wireless hotspots, thus creating a truly wireless solution as seen in Figure 3.2 below. The way this will work is that carriers would use rooftop base-stations, connected to the Internet. Each base station would make use of WiMAX technologies to connect to externally or internally mounted client-side antennas; thus allowing for both Non-Line Of Site (NLOS) and Line Of Site (LOS) connections.

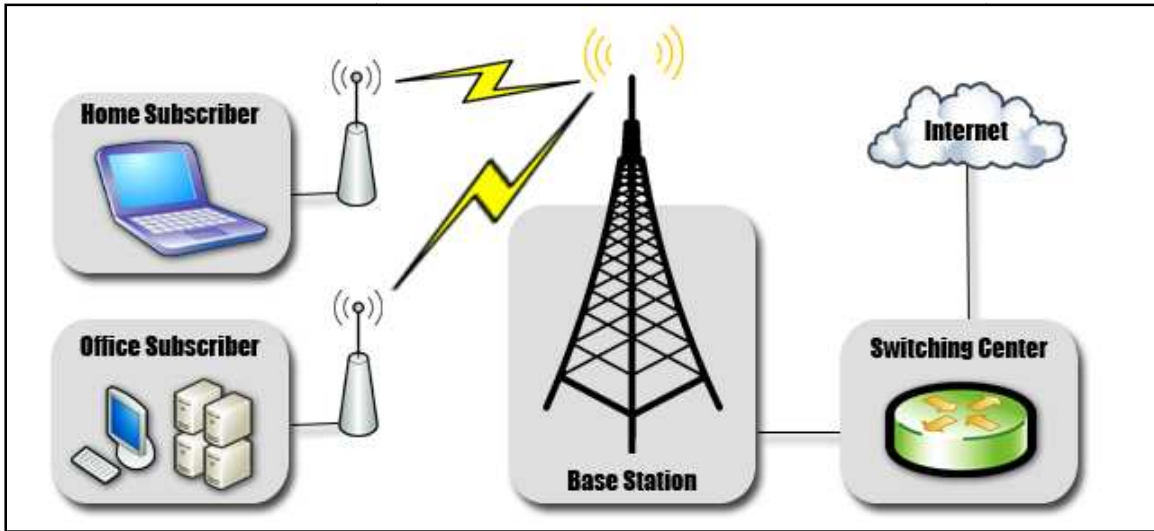


Figure 3.2: Simple WiMAX Implementation.

WiMAX, unlike most other wireless technologies, allows one to transmit data over multiple broad frequency ranges, allowing for multiple service providers to provide service over WiMAX and not clash over frequencies, as in 802.11 based networks. The 802.16d standard is set to run between the frequency bands of 2GHz and 11GHz. The standard was finalized in 2005, with amendments to the standard being finalized by late 2006 in the 802.16e standard. The new standard will support nomadic, mobile and stationary network solutions, running between 10GHz and 66GHz frequency bands (Motorola, 2005).

The name, WiMAX, has been loosely coined and is used to refer to both 802.16d and 802.16e standards. One of the problems that arises is that the 802.16d and 802.16e standards are not compatible. According to Motorola (2005), this fact will cause many organizations, which have already started implementing the 'd' standard, problems as they will have to replace all their equipment to gain the advantages of the 'e' standard. The reason for the incompatibility is that the 802.16d standard will make use of Frequency Division Duplexing (FDD), as opposed to the Time Division Duplexing (TDD) frequency plans of the 802.16e standard. Many companies are holding back on the installation of WiMAX until the finalization of the 'e' standard in late 2006. Security will more than likely become one of the talking points with WiMAX when it hits the market in earnest, especially noting the fact that WiMAX can also be mobile. Mobility on WiMAX is one of its key selling points: one can drive around and pickup signal and use the

network; but this, too, has huge security ramifications. In the next section, the focus is shifted to the different types of attacks that can occur on wireless and wired networks.

3.7 Hacking Wireless Networks

Now that a background has been provided on the generic network model, and the different standards based on the generic model have been discussed, it is now appropriate to look at the different attacks. Wireless networks, and the systems that support them, are vulnerable to a variety of attacks: both regular attacks that are also prevalent on their wired counterparts, but also a host of new attacks that specifically target wireless networks and cannot be run against regular wired networks. This section is dedicated to those attacks that specifically target wireless networks. As mentioned in a previous section, wireless networks pose their own problems due to the broadcast nature of wireless, and the fact that radio waves cannot easily be controlled or confined to a specific area.

There have been some attempts to combat attacks against organizational wireless networks and the dataflow in these networks, e.g., WEP (Wireless Equivalent Privacy), SSID (Service Set Identifier) hiding, etc., but these are merely irritations to a hacker trying to break into the network. According to McCullough (2004), WEP has two primary weaknesses that hackers exploit - its key distribution and the encryption, both of which have major flaws. Below are examples of three attacks that are based on the weaknesses described above.

Attack 1

WEP makes use of 40bit (weak) or 128bit (strong encryption) secret encryption key, but, the biggest problem with WEP is that a hacker needs to do nothing more than passively monitor the network and collect a maximum of 25GB of data. The hacker will then have the ability to get the encryption key and decipher the data flowing around the wireless network. Collecting this amount of data may take a few hours to a few days; depending on how busy the network may be (McCullough, 2004).

WEP has two methods of protection against the above passive sniffing attacks. The first is an IV (Initialization Vector), which is appended to the shared key, and is used to prevent the encryption of two cipher-texts with the same key stream. The second is the IC (Integrity Check) which, as the name suggests, is a field in the packet that makes sure that the packet stream has not been changed/modified during transmission of the data. Unfortunately, neither of these control mechanisms has been correctly implemented, thus resulting in poor security (Nichols & Lekkas, 2002).

Attack 2

Another attack resulting from the poor security and implementation of WEP is to actively inject traffic. In this attack, an attacker will know the plaintext of one message sent over the network, which he may or may not have sent, and with the encrypted version of this message, he will begin to construct correctly encrypted messages, which will be sent to the access-point or mobile device and will be accepted as valid. With some sifting of messages, an attacker may intercept SNMP messages and change them slightly to gain further access to an access-point (Borisov, Goldberg & Wagner, 2001).

Attack 3

A third attack that can be very dangerous to an organization is that of a Table-Based attack, where an attacker will build a table of all IVs (Initialization Vectors) and their corresponding key-streams. Around 15GB of storage space will be used, and all the hacker needs to get started is the plaintext of one single packet. This attack is possible because of the small number of IVs that can be generated before they are re-used. Once the hacker has his table built, he can decrypt any or all of the packets sent over the wireless network (Borisov, Goldberg & Wagner, 2001).

The three attacks listed above show how easy it is to actually attack a wireless network that has insufficient security mechanisms and weak encryption. If WEP encryption is the only option, the keys should be changed on a regular basis to avoid unauthorized people gaining access to organizational information and secrets. There are many other ingenious ways of attacking wireless networks. Through the problems associated with the WEP protocol, these attacks often expose the network to a host of generic hacks and attacks that can be run against either wired or wireless networks. Some of the generic attacks have been modified or mutated slightly to allow

for newer breeds that target wireless networks specifically. In the next section, many of the generic hacks and attacks used against organizational networks have been discussed.

3.8 Generic Hacks and Attacks

As concluded in the previous section, there are countless attacks and exploits that are available to a hacker wishing to infiltrate a network. All he/she needs do is scan the network and see which vulnerabilities have not been patched (McHugh et al., 2000). With Microsoft Windows, for example, estimates claim that over 94% of home-computer owners are running a version of MS Windows, and a large install base like this makes hackers interested in discovering vulnerabilities, to get “the most bang for their buck” as it were (McCullough, 2004). Most networks will have a few security holes in their systems, even though they are patched and up to date. At worst, a newly installed server may have no security patches installed at all, opening it up to a world of attacks (Broersma, 2006). This section shows some of the more prevalent types of attacks, which are generic to both wired and wireless networks, focusing, in some cases, on wireless-specific attacks, and how they have been modified to cause major problems for wireless network administrators.

There are generally three types of hackers, namely White Hats, Grey Hats and Black Hats (Hafner, 2000). The difference between them is in what they use their hacking skills for. On the one hand, white hats gain knowledge into hacking methodologies so that they can better combat the exploits that the black hats use to attack systems. Black hats are constantly striving to find new ways to damage organizational networks and computer systems, and mostly stick to the philosophy that information should be free according to Hafner (2000). Grey hats are in the middle: they often try to do some good in the world by posting hacking information, including hacks on the Internet, so that organizations can patch the holes. The problem comes in as many other black-hat hackers gain the same knowledge and use it in the reverse sense (Hafner, 2000).

These days, hackers are not just out to make names for themselves anymore. Now terrorists, known as cyber terrorists, are involved in the hacking game. The reasons that they use the electronic medium as their target are quite simple: they can remain anonymous, inflict huge

economic damage and, with the use of the Internet, can strike anywhere and anytime they wish. The psychological effects of cyber terrorism are the most effective part of their strategy: most people who bank and purchase online are fearful of having their money stolen online by someone who they cannot see (Weimann, 2004). Attacks have been evolving over the past twenty or so years, from attacks aimed at single computers in the 1980s to individual networks being targeted in the 1990s. Lately, it has emerged that the global computing infrastructure is a prime candidate for attack. This can be seen by the attack that occurred in October, 2002, where an attack was launched against the 13 Internet root servers providing the Internet's core DNS, effectively cutting the Internet off for about an hour (Ciampa, 2005; Weimann, 2004).

Some of the most menacing attacks are listed below, and these are performed against both wired and wireless networks. Many of these attacks are more effective in a wireless environment because of the ease of connection into the network in the absence of wires.

- 1. Man-in-the-Middle Attack:** Man-in-the-middle attacks are interception attacks, whereby an attacker will intercept data destined for a valid user, modify the information contained in the data transmission and send the newly modified information on to the recipient, as can be seen in Figure 3.3 (Ciampa, 2005; Boyd & Dasgupta, 2004). An example of how this attack could take place would be if a hacker places a fake website on a corporate LAN, which looks like an Internet banking site for instance, and diverts access from an actual bank site to his fake site. Users will browse to what they think is the bank site and will type in all their information, including pin numbers. The fake site will then redirect them to the real bank site, and the users will be none the wiser that they have had their information stolen. Within a wireless environment, this attack is easier to perpetrate than in a standard wired LAN, because a hacker does not need to be connected directly into the network by any wire, he can merely sniff packets off the radio waves and modify them, as was discussed in Section 3.5.

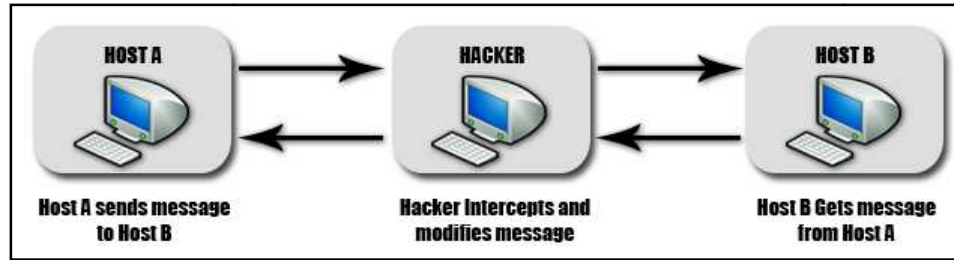


Figure 3.3: Man-in-the-Middle Attack

Man-in-the-middle attacks can be either active or passive in nature. With passive attacks, the hacker will capture sensitive data and send it on without his presence being noticed, whereas in an active attack, the contents of the captured information is altered, thereby greatly increasing his chance of being noticed (Ciampa, 2005).

2. **Replay Attack:** A replay attack is similar in nature to the man-in-the-middle attack, discussed in the previous point, in many respects. However, replay attacks capture the message travelling between hosts, and it is stored and sent again later when the hacker “replays” the original message. The point behind this attack is to gain a trusted relationship between attacker and a server, e.g., when the hacker intercepts maintenance messages or requests between a fileserver and active directory server, he can store the message and later, at will, replay the message to the active directory server and thus gain a trusted link to it.

These messages are usually encrypted. By gaining the server’s trust, the attacker can gradually change the message bit by bit, and in doing so, eventually works out the contents of the encrypted message by the replies received from the active directory server (Ciampa, 2005). The process is shown in Figure 3.4. This attack could be used against an access-point to gain access to its internal configuration, by replaying administrative SNMP requests to the access-point from network management applications. From here, the attacker may gain valuable information, such as MAC addresses of clients and WEP keys.

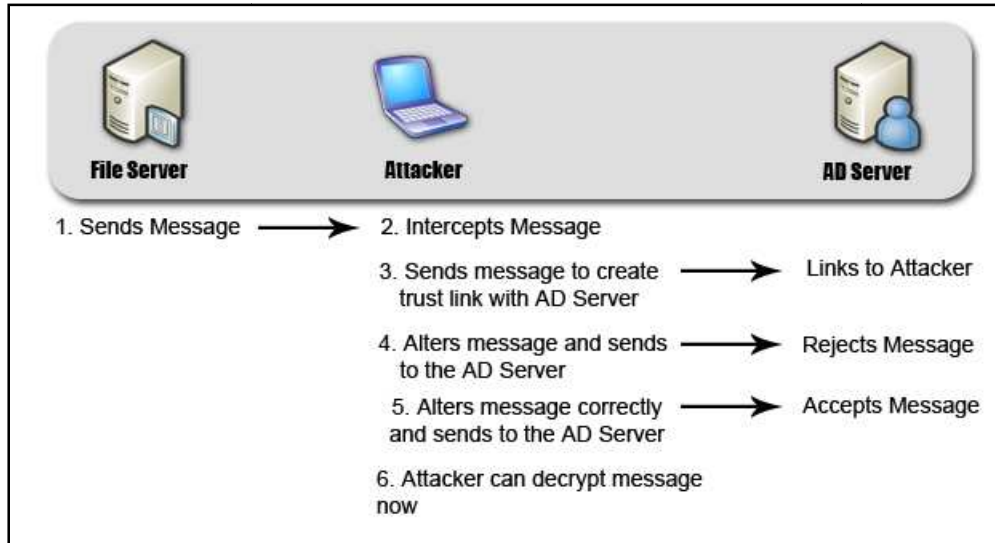


Figure 3.4: Replay Attack

3. Denial of Service (DOS) Attacks:

Denial of service attacks can be one of the most devastating attacks against a network, in that this kind of attack denies legitimate users access to resources that they may need to be productive in their jobs. Some of the ways DOS attacks can be performed on a network are listed below (CERT Coordination Center, 2005):

1. Attempts to “flood” a network, thereby preventing access.
2. Attempts to break communication between machines in the network, thereby stopping flow of data.
3. Attempts to prevent individual users from accessing information or services.
4. Attempts to disrupt services to a specific system or user.

One of the most common types of DOS attacks run against an organization is that of bandwidth consumption. When an organization’s bandwidth is consumed by a DOS attack, the organization not only loses revenues from lost e-mails, e-commerce sales and the likes; but also loses the capital spent on the actual bandwidth itself, which can run into tens of thousands of Rands per month. To run a bandwidth-intensive DOS attack, an attacker will infect many thousands of computers of unsuspecting Internet users with an

application. The infected computers form what is known as a bot-net, a network of bots that wait for the master's command to start flooding a specified website or organization.

The bots, also known as zombies, do nothing on the machine which they infect until told to do so by the bot-master/hacker. This makes them hard to detect (Kawamoto, 2005). Most zombie-based DOS attacks are started via IRC (Internet Relay Chat), as the zombie software infecting unsuspecting user machines has code contained within it that makes a connection to a specific chat room on a server. When the bot-master types in a specific command; the attack commences via all bots simultaneously.

4. Distributed Denial of Service (DDoS) Attacks:

A variation of the DOS attack mentioned in the previous section is called DDoS or Distributed Denial of Service. These attacks do not require the attacker to infiltrate the target network in any manner. This, coupled with the fact that the attack is easily perpetrated, makes it a favourite with hackers in that it is extremely difficult, if not impossible, to determine the origin of the attacker (Jakobsson & Menczer, 2005).

The way a DDoS attack works is quite simple and to a point, similar to the standard DOS attack. The attacker will spend some time at the beginning "recruiting" or hacking into un-patched computers and will infect them with the attack software. The difference between this and a regular DOS attack is that the attack software injected into the zombie machines can further recruit other machines that they find in their directory service, or any machine they can get access to over the Internet or LAN. This is what makes the DDoS attack so problematic and virtually untraceable. Most attackers, when perpetrating a DDoS attack, will spoof their IP and MAC addresses to further mask who they are and where they are situated (Mirkovic, Martin & Reiher, 2002). This kind of DOS attack utilizes IRC as the means to invoke the attack.

This kind of attack can be used via a wired or wireless network to attack a server or the entire organization's Internet pipe. DDoS attacks can also have a detrimental effect on a wireless network. In a wireless-specific DDoS attack, known as jamming, any infected wireless device sends junk data to the access point, and jams signals to valid users (Boyd

& Dasgupta, 2004). Jamming can be used as a DOS or DDoS attack and can cause major congestion problems on the targeted wireless network.

5. Fake/Rogue Access-Point Attacks:

Many organizations, such as coffee shops, and wireless broadband providers have login and passwords for their wireless networks to allow users, who have paid their subscription, access and to keep others out. There is an attack that hackers have developed that allows one to steal usernames and passwords by creating a fake access-point that users will connect to, thinking that it is the access-point of the service provider, because the intruder's access-point has the same SSID as the legitimate service provider's one (Boyd & Dasgupta, 2004).

The intruder may even allow the user to connect through his access-point to the coffee shop's access-point, and the user will not be any the wiser that his/her credentials have been stolen. The attack can go one step further in that the attacker can now perform a man-in-the-middle attack against the user for further information. By setting up a fake/rogue access-point, the attacker might catch many users at once as they stray into his wireless signal.

After looking through the above attacks, one gets a sense of what lengths a hacker will go to in order to gain access to a network. Hackers' motives, as stated previously, range from creating a name for themselves in the hacker community, to cyber terrorism causing panic in the public domain. Part of this panic is the fear of organizations and individuals of using network infrastructure in case they get targeted. The attacks mentioned in this section can affect an organization, no matter whether it uses a wired or wireless-based network. Organizations and the public need to be aware of the dangers and protect themselves by implementing security policies and security mechanisms.

3.9 Conclusion

With the number of attacks increasing on an annual basis, and the lack of regard for public laws that hackers have, organizations need to implement strict security protocols and install and keep their security mechanisms, such as intrusion detection systems, up to date. Gone are the days that an organization can just install a new technology and hope that no-one will find a security hole, especially in the case of wireless-based networks, as was discussed earlier. These networks are inherently insecure and should have systems in place to monitor and prevent attacks in the best scenario, before the attacker actually fully initializes his attack against the organization.

As was discussed previously, the history of networking has been full of innovation and technological breakthroughs. This allows organizations to operate more efficiently and create greater opportunity for profit. The problem is that as the speed of networks grows, so does the possibility for attack. Section 3.8 discussed some of the attacks currently being run against networks. This information can be used to learn how a hacker thinks and will allow administrators to understand where and how future attacks may occur. In the next chapter, Wireless Intrusion Detection Systems (IDSs) are discussed, including currently available Wireless IDSs and their features. A generic attack taxonomy has also been discussed, allowing one to understand both generic wired and wireless attacks.

Chapter 4

Wireless Intrusion Detection Taxonomy and Products

4.1 Introduction

IDSs, as highlighted in a previous chapter, are software systems designed to detect and prevent the misuse of computer networks (Aickelin et al., 2004). Intrusion Detection (ID) and IDSs have come a long way since their beginnings in the early 1980s with John Anderson and Dorothy Denning. In the event that an organization is faced with a security breach, including the theft of valuable information or even damage to mission critical computer systems, today's IDSs are able to answer many questions in any environment in which they may be installed. Some of these questions are listed below (McHugh et al., 2000):

1. What actually happened?
2. Who was/is affected and how?
3. Who is the intruder?
4. From where and when did the attack originate?
5. How did the intrusion occur?
6. Why did the intrusion occur?

Answers to these questions may be a good starting point in finding and bringing the intruder to justice and, in some cases, allowing an organization to possibly regain some lost reputation. IDSs are also important because computer systems and information technology infrastructures within organizations are becoming so complex that it is virtually impossible for a single individual to understand, or think of administering them in a secure and sound manner (McHugh et al., 2000). With the help of IDSs, system administrators are only given the information needed to do their jobs effectively, while the IDS sifts out the information that is not a necessity

for him/her to know. There are, according to Petrovic (2005), two basic assumptions that relate to a successful IDS/Intrusion Prevention System (IPS) and they are as follows:

1. All system activities are observable.
2. Both normal and intrusive behaviour/activities have distinct evidence.

IDSs must take these assumptions into account to allow them to fully discover where, when, who and how the attacker performed his attack. As will be discussed, IDSs are not miracle cures for organizational security. They, too, have their flaws and shortcomings. That is why one has to understand how intrusion attacks are perpetrated, how to categorize these attacks, as well as how to classify the dynamic nature of intrusion attacks. This chapter will attempt to address these issues by investigating currently available intrusion taxonomies and the development of a Proactive Generic Intrusion Taxonomy.

4.2 Intrusion Taxonomies

As discussed in Chapter 3, within the world of information technology today, there are many ways in which an attacker can disrupt or cripple an organization's computer systems and their communications networks. Although one may not always know the reasons or motives behind an attack, it is vital that one does a forensic investigation in order to prevent new or future attacks. One way an organization can do a forensic investigation is to classify an attack and compare its characteristics to other known attacks with similar characteristics.

This classification of an intrusion attack is known as an intrusion taxonomy and is defined as "the study of the general principals of scientific classification" (Alessandri, 2001). According to Axelsson (2000), a taxonomy serves three purposes: **Description**, **Prediction** and **Explanation**, the definitions of which are listed below.

1. **Description:** A taxonomy helps us describe something scientifically, and provides one with a tool with which to order the complex phenomena that surround it into more manageable units.
2. **Prediction:** By classifying a number of objects according to a taxonomy and then observing the 'holes' where objects may be missing, one can exploit the predictive

qualities of a good taxonomy. Good taxonomies could point one in the right direction when undertaking further studies.

3. **Explanation:** A good taxonomy will provide one with clues about how to explain observed phenomena.

The next few subsections give a background into intrusion taxonomies, allowing one to understand the need for a very generic intrusion taxonomy that allows for multiplatform environments. The background includes a few important attack taxonomies relevant to intrusion attacks and serves as reference for defining a **proactive intrusion taxonomy**. This proactive intrusion taxonomy not only allows one to understand existing attacks, but it can also be used to proactively prevent new attacks. These attack taxonomies are listed in the subsections to follow.

4.2.1 Bishop's Vulnerability Taxonomy

Bishop's Taxonomy of software vulnerabilities is a six-axis taxonomy, where each axis contains a vulnerability classification (Vijayaraghavan, 2003). The axes, as described by this taxonomy, are listed below (Du & Mathur, 1997):

1. The nature of the flaw;
2. The time of introduction;
3. The exploitation domain of the vulnerability;
4. The effect domain;
5. The source of identification of the vulnerability; and
6. The minimum number of components needed to exploit the vulnerability.

Bishop's Taxonomy describes vulnerabilities in a form that is useful to Intrusion Detection mechanisms and primarily deals with vulnerabilities in a UNIX environment (Carver & Pooch, 2000).

This taxonomy may be useful to an IDS in determining where the flaw lies in the system, when the attack took place and how far the effects of the attack spread over the system before it was halted by a security mechanism, or the attack halted itself. The problem with this taxonomy is the fact that it only shows the classification of vulnerability

primarily in a Unix environment. This does not help an IDS operating on an MS Windows platform.

Another problem with this taxonomy is that it does not allow for the proactive identification of attacks as they occur on a system, e.g., it does not specify how an attacker can actually take advantage of the vulnerability and how best to detect this form of attack. The IDS can only use Bishop's Taxonomy after an attack has occurred to classify the attack in terms of the axes listed above and, in so doing, forensically trace back the attack, hopefully finding the attacker.

4.2.2 Aslam's Taxonomy

This taxonomy was constructed to categorize attack and vulnerability data stored in a database. This taxonomy is very detailed, but one problem is that it only considers Unix-based vulnerabilities and attacks in its implementation level (Bishop & Bailey, 1996). The taxonomy classifies coding errors into two categories: synchronization errors and condition validation errors.

The taxonomy attributes all non-synchronization security errors to the improper evaluation of condition (Du & Mathur, 1997). Simply, this means that one can fix the error without even changing any condition in the application. Below is the implementation level of Aslam's Taxonomy (Bishop & Bailey, 1996; Vijayaraghavan, 2003):

1. Operational Fault (Configuration Error):
 - 1.1 Object installed with incorrect permissions;
 - 1.2 Utility installed in the wrong place;
 - 1.3 Utility installed with incorrect setup parameters.
2. Environmental Fault;
3. Coding Fault:
 - 3.1 Condition validation error:
 - 3.1a. Failure to handle exceptions;
 - 3.1b. Input validation error;

- 3.1b.i. Field value correlation error;
 - 3.1b.ii. Syntax error;
 - 3.1b.iii. Type and number of input fields;
 - 3.1b.iv. Missing input;
 - 3.1b.v. Extraneous input.
- 3.1c. Origin validation error;
- 3.1d. Access rights validation error;
- 3.1e. Boundary condition error.
- 3.2 Synchronization error:
 - 3.2a. Improper or inadequate serialization error;
 - 3.2b. Race condition error.

Aslam's Taxonomy is quite detailed because of its database of vulnerability and allows an intrusion detection system the ability for more in-depth classification of implementation-level flaws.

As with Bishop's Taxonomy, Aslam's Taxonomy only takes Unix-based attacks into account and, therefore, is not exhaustive. As stated before, this causes problems for an IDS implementing this taxonomy when attempting to use it in the classification of MS Windows-based platforms.

Although this taxonomy allows for depth while classifying vulnerabilities at implementation level, it lacks high-level categories to classify design flaws (Bishop & Bailey, 1996). This taxonomy is also very ambiguous in that it allows one single vulnerability to be classified into many categories (Bishop & Bailey, 1996).

4.2.3 Neumann & Parker's Taxonomy

The Neumann & Parker Taxonomy is a taxonomy based on empirical data used to classify actual attacks. The empirical data that the taxonomy is based on was collected by Neumann at SRI International as part of their Risk's Forum "Risks to the public in

computers and related systems” (Howard, 1997). The taxonomy contains eight categories into which an intrusion can be classified, and these are listed below in Table 4.1 (Botha, 2003; Howard, 1997).

CLASSES	INCIDENTS	RELATED ACTIONS
NP1	External Misuse	Non-technical, Physically separate intrusions
NP2	Hardware Misuse	Passive or active hardware security problems
NP3	Masquerading	Spoofs and Identity changes
NP4	Subsequent Misuse	Setting up intrusion via plants, bugs etc.
NP5	Control Bypass	Circumventing authorized protections / controls
NP6	Active Resource Misuse	Unauthorized changing of resources
NP7	Passive Resource Misuse	Unauthorized reading of resources
NP8	Indirect Aid	Neglect or failure to protect a resource
NP9	Indirect Aid	Planning tools for misuse

Table 4.1: SRI Neumann & Parker (NP) Taxonomy

Implementation of Neumann & Parker’s Taxonomy provides a wide range of incidents and incident types that include hardware and software misuse. This provides a good base to perform ID from. This is because most attacks consist of not only software misuse, but all kinds of misuse activity.

The problem with Neumann & Parker's Taxonomy is the same as most vulnerability taxonomies, in that it has been designed at a higher level of representation to be of any real help in actual intrusion detection (Axelsson, 2000).

4.2.4 Lindqvist & Jonsson's Intrusion Taxonomy

Lindqvist and Jonsson defined two taxonomies that differ from most taxonomies up to this time, in that their schemes categorized intrusion attacks based on the result of an actual attack, and what method or technique was used in the attack (Paulauskas & Garsva, 2006; Carver & Pooch, 2000). The taxonomy has three main objectives and these are listed below (Carver & Pooch, 2000).

1. Establish a framework for the systematic study of computer/network attacks;
2. Establish a structure to report computer attacks to the incident response team;
3. Provide a mechanism for determining severity of attacks.

The taxonomy is based on work done previously by Peter Neumann and Donn Parker and provides an extension of their work (Vijayaraghavan, 2003). The taxonomy expands three of the Neumann and Parker categories, namely **Bypass**, **Active Misuse** and **Passive Misuse** (Vijayaraghavan, 2003). See Figure 4.1 for the full taxonomy (Carver & Pooch, 2000).

As this taxonomy is an extension of the one done by Neumann & Parker, it should be noted that it has the same problem of being designed to a higher level of representation than to be used actively in actual intrusion detection. This taxonomy also has a problem with the fact that it only deals with the classification of a vulnerability once it has already occurred. By the time one has found the attack and its details the attacker is long gone.

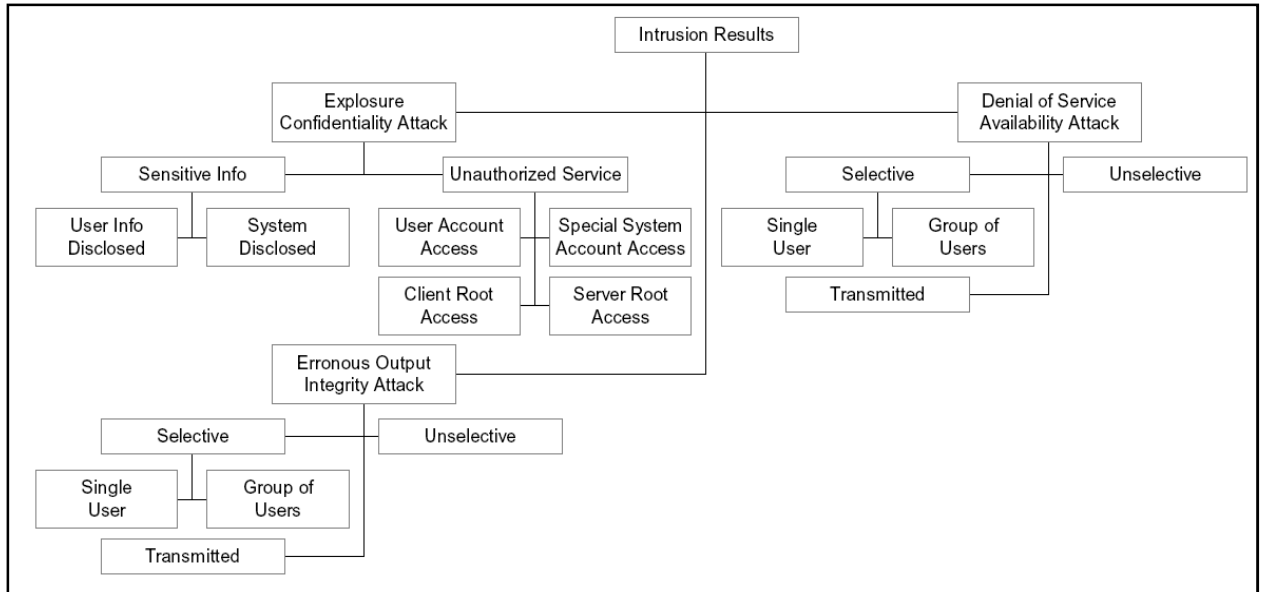


Figure 4.1: Lindqvist & Jonsson's Taxonomy

4.2.5 Landwehr's Taxonomy

Landwehr's Taxonomy is slightly different to most intrusion and vulnerability intrusions in that it uses three dimensions to classify an intrusion / vulnerability instead of the single-dimension schemes of other taxonomies (Du & Mathur, 1997). The three dimensions used by this scheme are **Genesis**, **Time of Introduction** and **Location** (Du & Mathur, 1997; Carver & Pooch, 2000). Genesis refers to how a flaw finds its way into an application or program. Time of Introduction refers to the point in the software development life cycle where the vulnerability or flaw is introduced. Location refers to the part of the operating system, software application or hardware where the flaw lies.

With the Genesis, it is possible to avoid, detect or compensate for security flaws according to Du & Mathur (1997). The time of introduction is important and allows one to see in which phase of software development the flaw was introduced and thereby helps to strengthen the software development process itself. The Location variable is also important in that it allows one to see where the flaw was introduced in the system. This helps an organization to better protect itself against flaws (Carver & Pooch, 2000).

Landwehr's Taxonomy does not allow for proactive detection of attacks but instead provides a way to detect attacks that have occurred accurately using the Genesis, Time of Introduction and Location methods.

As with the previous taxonomies, Landwehr's Taxonomy has some shortcomings. The main criticism has to do with the ambiguities in the Time of Introduction category (Lough, 2001).

4.3 Proactive Generic Intrusion Taxonomy

The previous section highlighted some of the more popular and well-known intrusion and vulnerability taxonomies. Most of these taxonomies deal with attacks that have already happened and show reports of attacks that occur hourly, daily, weekly etc. The ability to detect an attack after it has taken place, via audit logs, etc., is good to have; but it stands to reason that if an attack can be detected before the attacker has actually finalized his/her attack, this would be far better in terms of organizational security. Stopping an attack before its final payload has been dropped would enable the organization to theoretically better protect its information assets from the following consequences (Botha, 2003):

- Theft of organizational information;
- Corruption of organizational information;
- Disclosure of organizational information and corporate secrets;
- Denial-of-service to valid users.

For the purposes of this dissertation, an intrusion attack refers mostly to a sequence of intrusion actions that cause one of the aforementioned consequences. Later in this section, research done at the Nelson Mandela Metropolitan University (NMMU) on a Generic Proactive Intrusion Taxonomy is discussed, and additions to the taxonomy have been made to bring it to a point in which it can be used with wireless-based attacks.

According to the findings of the research paper, one has to revise the two main pillars of Intrusion Detection: **Pillar 1** where an intruder must perform a sequence of related actions and **Pillar 2** where the intruder must utilize a set of resources. The revision of the pillars must be done in order to consider a proactive intrusion taxonomy. It follows that an IDS that implements a generic and proactive taxonomy should have the ability to detect most attacks thrown at it and, in a few cases, this may occur in a reactive manner (Botha, 2003). As mentioned before, in some cases, the IDS should detect intrusions not yet seen.

Botha's Proactive Generic Intrusion Taxonomy, seen in Figure 4.2 below, takes into account the two pillars mentioned before and pays special attention to the first pillar.

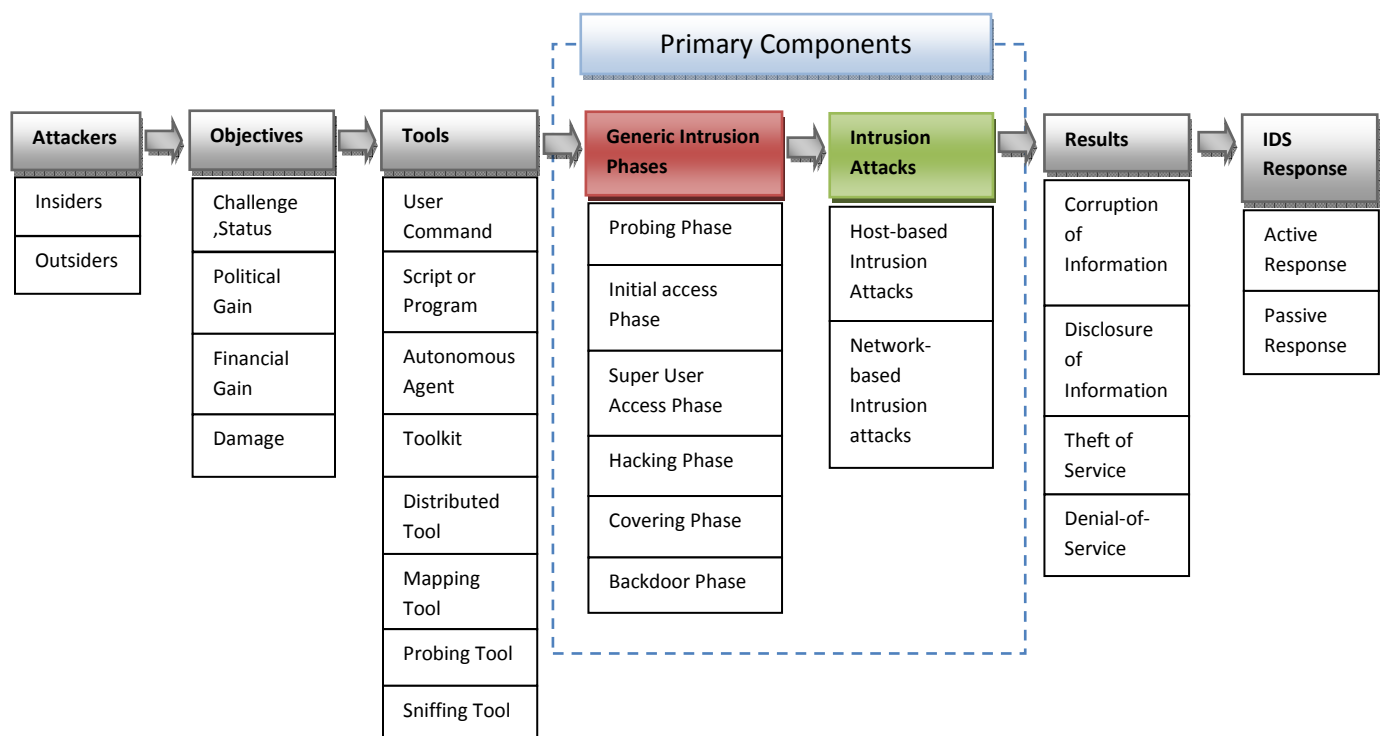


Figure 4.2: Proactive Generic Intrusion Taxonomy (Botha, 2003)

The Figure 4.2 taxonomy consists of seven components, including the two main components: **Generic Intrusion Phases** and **Intrusion Attacks**. The Generic Intrusion phases consist of six phases identified by Botha (2003) that describe the steps a hacker/intruder must go through in order to complete an attack on a system. During this research, the writer clearly indicated that

the intruder can perform the six steps in a different order, or he/she can opt to skip one or two of the steps. In general, however, research shows that at least four of the steps should be performed by the attacker (Botha, 2003). The generic intrusion phases are listed below including descriptions of what each phase focuses on (Botha, 2003).

- 1. Probing Phase:** The intruder will gather information about the organization and its users that he will be targeting. Hackers external to the organization will spend a great deal more time in this phase than internal users, as they may not know the organization as well. The attacker then creates his plan of how, when and what tools and access methods he will use to perform his attack. Probing or scanning the organization's services can also be a good method to finalize the intrusion plan.
- 2. Initial Access Phase:** Internal attackers need not complete this phase, as they already have access to the resources and systems needed in the attack. External hackers will need to try to gain access to the system by identifying holes in the organization's security. This could be finding bad password entries or gaining basic rights without required authentication.
- 3. Super-User Access Phase:** All intruders need to gain super-user access or full access rights to the system in order to perform most intrusion attacks. This can be achieved by acquiring an administrative password or by exploiting vulnerabilities in either hardware or software.
- 4. Hacking Phase:** The hacking phase is where the attacker will perform his actual intrusion into the system. The actions he performs in this phase range from deleting files to changing system configurations. In some extreme cases, where the attacker wishes to cause grievous damage to the organization, he/she will crash the system via use of denial of service attacks.
- 5. Covering Phase:** After an intruder has finished his/her attack, he/she will then attempt to erase all traces of their activities on the system. This is done before the system administrator actually realizes that an attack has taken place. Most intruders remove traces of themselves by using tools that edit audit logs instead of deleting them, as this would raise alarms. If the attacker does not edit the logs, but rather deletes the files, then the chance of the administrator detecting the attack is high.

6. **Backdoor Phase:** After the intruder has erased all traces of him/herself from the system, the next step is to place software tools on the system that will allow the attacker to access the system at a future date. Some of these tools allow super-user access when a special password is entered into the logon screen, which is part of the backdoor application.

The second main component of Botha's Generic Intrusion Taxonomy is that of **Intrusion Attacks**. There are two parts to this component. Both are listed below, with explanations of their workings.

1. **Host-Based Intrusion Attacks:** Host-based intrusion attacks are attacks that are aimed at a single host on a network.
2. **Network-Based Intrusion Attacks:** Network-based intrusion attacks are attacks aimed at an entire network. An example of a network-based attack is a DoS attack.

The author of the paper did not go into depth about wireless networks, in part because at the time the paper was written, there was not as much demand for organizational wireless networks; thus, the need for wireless security was not a priority. With current demand ever increasing for wireless segments to existing networks, the need to secure the networks is gaining priority.

Although wireless networks are similar to regular wired networks, they have vulnerabilities that are unique to wireless networks that do not affect their wired counterparts, e.g., WEP attacks and Fake Access-Points. For this reason there needs to be a third part to the Intrusion Attacks component of the Proactive Generic Intrusion Taxonomy. This third part can be seen in Figure 4.3. as **Wireless Network-Based Intrusion**. This component updates the taxonomy, allowing it to take into account the wireless networks that most organizations are implementing currently. The updated taxonomy allows an IDS implementing it to protect an organization's networks from attacks originating on both wired, wireless networks and on the hosts themselves.

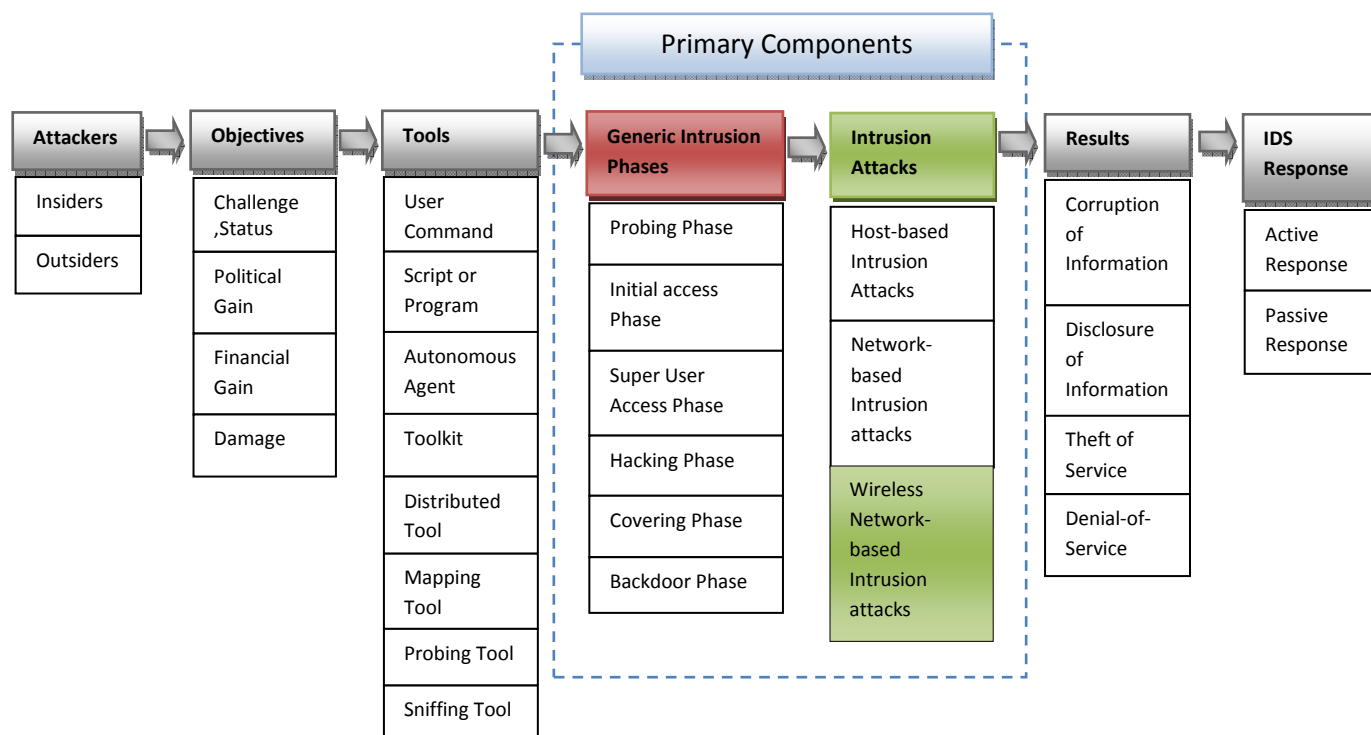


Figure 4.3: *Updated Proactive Generic Intrusion Taxonomy*

The next section discusses currently available IDS products, research, commercial and public domain software and hardware. This provides a background into what IDSs currently can do when it comes to ID and prevention. The focus is on whether any products currently available, implement wireless ID components, and if so, how well these components work.

4.4 Commercial, Research and Public Domain IDSs

In the world of IDSs, there are three categories of IDSs that are available: commercial, research and public domain IDS. For the purposes of this research, Commercial IDSs are thought of as IDS products aimed at the mass market, usually closed source and have support offered by the software developer. Research IDSs on the other hand are classified as purely for research purposes and are not for commercial sale or use. Public Domain IDSs are grouped by the fact that the software is free of charge to anyone who wishes to use it. It also has licensing agreements differing to commercial licences, support for these products is usually offered by a

host of companies and not necessarily the software developers. Public Domain IDSs are most commonly also open-source software. This section lists and outlines the features and benefits of some of the most common IDS products. The IDSs have been listed and grouped by the categories listed earlier, within each category there is an explanation of the various IDSs available in the category. This includes features, IDS structure and how well the IDS performs its tasks.

4.4.1 Public Domain IDSs

4.4.1.1 SNORT: SNORT is an open-source/public domain IDS and can be installed on a multitude of Linux and Unix platforms. SNORT is a NIDS and can perform real-time analysis of network traffic (McHugh et al., 2000). SNORT allows for plug-ins created by other open-source groups, allowing for a scalable and highly customizable IDS according to the project's website (www.snort.org). SNORT has the ability to detect attacks such as (McHugh et al., 2000):

- Buffer overflow attacks;
- Stealth port scans;
- CGI-based attacks;
- SMB probes.

Because SNORT is open-source and public domain, there are many developers contributing to the IDS and it is under rapid development. SNORT is a popular IDS as it is free for anyone to use and has regular updates, with the many side projects adding to the functionality, including SNORT log analyzers (McHugh et al., 2000).

With the addition of a separate plug-in project called SNORT-Wireless, SNORT IDS can gain the ability to detect 802.11 wireless intrusions (Lockhart, 2005). SNORT-Wireless allows the SNORT IDS the ability to detect the following (Lockhart, 2005):

- Netstumbler;
- Rogue AP;
- AdHoc Network.

The problem with this add-on project is that the user has to have a good knowledge of how to compile Linux projects. The project also only currently caters for Linux/Unix environments; thus, not allowing for security on mixed platform networks.

4.4.1.2 SHADOW IDS: This IDS was developed to run on inexpensive computer hardware running open-source, public domain or freely available operating systems and software (Shadow Team, 2003). The SHADOW IDS has two parts: the first is a sensor located near the organization's firewall and the second is an analyzer inside the firewall itself (Shadow Team, 2003). SHADOW IDS performs packet analysis on all packets entering the monitored network. Software packages tcpdump and libpcap are used for this purpose (Shadow Team, 2003).

The network sensors search through the tcp headers and search for information of interest. The analyzer then analyzes all the information and outputs it to a webpage containing all alerts. Shadow IDS currently has no wireless specific detection capabilities and with the current growth in popularity of wireless based networks, this detracts from the IDS as opposed to other IDSs with the ability to detect wireless attacks.

4.4.2 Commercial IDSs

4.4.2.1 Internet Security Systems (ISS) RealSecure: This software-based IDS works on a three-part system: network-based detection engine, host-based detection engine and an administrator's module (McHugh et al., 2000). RealSecure provides response to both network and host-based intrusions, by blocking IP addresses, ports, etc., on the network and by locking user accounts and termination of user processes on the host side (McHugh et al., 2000). The RealSecure product range has capabilities to monitor both 100Mbit and 1Gbit networks (ISS, 2006). This product can provide intrusion detection on multiple platforms such as Linux, Solaris, Windows and IPSO (ISS, 2006). Some of the features of RealSecure have been listed below (ISS, 2006).

- Cutting-edge accuracy and performance;
- Advanced event correlation;
- Ease of deployment;
- Ease of maintenance.

RealSecure has recently been updated to allow for the detection of wireless attacks. The new system, called “RealSecure Protection System”, is deployed between the wireless Access-Point and the corporate network (ISS, 2001). The system relies on Simple Network Management Protocol (SNMP) to gather information from the hosts and wireless devices as well as sniffing wireless traffic for problems. The wireless scanner constantly searches for misconfigured devices and identifies unauthorized devices.

Although ISS is currently the leader in commercial IDS technology (ISS, 2002), its focus is mainly on wired network segments and not on the wireless segments. The RealSecure product only detects some of the wireless attacks and vulnerabilities, according to their product documentation. This is a problem as it may lull an organization into a false sense of security (ISS, 2002).

4.4.2.2 NFR (Network Flight Recorder) Sentivist: This IDS is a hardware-based IDS/IPS solution and is actually an IPS system according to the manufacturer NFR (2006). The IPS is in its fifth version now and also contains a firewall as part of its prevention approach (NFR, 2006). The NFR product claims to stop the following attacks before they can cause damage to one’s network (NFR, 2006):

- Automated malware;
- Information theft;
- DoS and DDoS;
- Command tampering;
- Existing vulnerabilities;
- Unsanctioned network changes.

The NFR Sentivist also has quite a few benefits and features that have been listed below (NFR, 2006).

- Real-time threat detection and prevention;
- Protection beyond IDS / IPS with firewall capabilities;
- Extreme usability;
- Network node intelligence information;
- Situational awareness and control.

NFR's Sentivist is extremely expensive and does not have a wireless intrusion detection component. Rather NFR focuses on large-scale corporate wired network infrastructure containing hundreds or thousands of Sentivist sensors, costing between \$11 000 and \$20 000 (Foster, 2006).

4.4.2.3 Symantec NetProwler: Symantec's NetProwler allows for the instant detection, logging and termination of misuse, abuse or corruption of computer networks by both internal and external attackers (Symantec, 2006a). This hardware-based NIDS uses a Stateful Dynamic Signature Inspection (SDSI) engine, patented by Symantec. The device is said to be able to stop even the most sophisticated attacks via its attack definition wizard and the SDSI, allowing administrators to protect their corporate resources from hundreds of known attacks and new unseen attacks (Symantec, 2006a). Below can be found a list of some features of NetProwler (Symantec, 2006a).

- Network Profiling for "out-of-the-box" installation and automatic configuration;
- Comprehensive attack signature customization wizard to protect company-specific applications;
- On-the-fly loading of updates and new attack signatures while keeping defenses on-line and current;
- Integration with AXENT's award-winning Intruder Alert™ for enterprise monitoring of network and host security events.

Symantec's NetProwler does not currently have any wireless capabilities. The latest version is 3.5, and it looks like the product will not be continued after this release, so it is doubtful that NetProwler will be updated to gain wireless detection capabilities.

4.4.2.4 Tripwire: This software-based product independently audits changes across the entire organization, including servers, desktops, databases and application programs (Tripwire, 2006). Tripwire has the ability to audit a wide range of network products for change, including switches, routers, firewalls, VPN concentrators and network storage devices (Tripwire, 2006). When a change occurs Tripwire logs it and reconciles it against a list of authorized changes. If the change was not sanctioned it is flagged as a possible attack (McHugh et al., 2000). A database also exists with an exhaustive list of changes,

who made them, whether or not they were sanctioned. Tripwire can be very difficult to setup, especially in large installations with multiple sites, as it may be difficult to determine which files may be associated to services and should be allowed to change (McHugh et al., 2000). Some of the features of Tripwire are listed below (Tripwire, 2006).

- Multivendor infrastructure coverage;
- Multiple levels of change detail;
- Event driven and periodic change detection;
- Archived audit trails;
- Comprehensive reporting and dashboards.

Tripwire has not currently got any wireless intrusion-specific detection components available.

4.4.2.5 Cisco Secure IDS: This product was formally known as Cisco NetRanger. The Cisco product is actually a device that has been specifically designed for large corporate / service provider networks and is available in a variety of sizes, scalable to organizational needs (Cisco Systems, 2006). This product, being a physical device, is able to handle vast quantities of traffic and can accurately detect attacks. The device is able to forward alarms to regional, national or international headquarters and has the ability to integrate with Cisco Catalyst 6000 based switches (Cisco Systems, 2006). Some of the key features are listed below (Cisco Systems, 2006):

- Pervasive platform support;
- Scalable sensing performance;
- Investment protection;
- Active response;
- Transparent operation;
- Sophisticated attack detection and anti-hacking protection.

Cisco's Secure IDS does not directly have a wireless IDS component. The company does, however, have a separate wireless IDS, which requires one to implement Cisco's WLSE (Wireless LAN Services Engine) and also requires the use of Cisco Access-Points only, so

this is a very proprietary system and not good for organizations that have already got wireless systems in place.

4.4.3 Research IDSs

4.4.3.1 Emerald: Work into this research IDS began in 1983 with the groundbreaking work done on an algorithm called the Multivariate Statistical Algorithm. This algorithm allowed for characterization of user behaviour (McHugh et al., 2000). The name Emerald is an acronym for Event Monitoring Enabling Responses to Anomalous Live Disturbances. The goal of the Emerald research was to provide ID to large distributed organizational networks that are loosely coupled (McHugh et al., 2000). These distributed networks are harder to monitor than single-site networks, as infrastructure can be distributed worldwide, and network connection speeds vary. Emerald contains engines for both signature and anomaly detection and is said to be the way IDSs will be headed in the future (SRI, 2006; McHugh et al., 2000). Some of the features of the Emerald research IDS are listed below (SRI, 2006).

- Scalable network surveillance;
- High-volume network analysis;
- Light-weight distributed network sensors;
- Generic infrastructure.

As with Tripwire and some other IDSs, Emerald does not contain any wireless intrusion-specific detection components.

4.4.3.2 STAT: STAT (State Transition Analysis Technique) is another research IDS developed at the University of California in Santa Barbara. The theory behind STAT is that each attack can be represented as a series of actions, and these actions together form the attack (McHugh et al., 2000). Each attack is represented in a graphic notation (STATL), done by state transition notation and precisely identifies the requirements of the attack, as well as the nature of the attack (McHugh et al., 2000). This is a signature-based IDS and signatures are abstractions of an attack scenario. Each signature can detect a whole family of attacks, including never before seen variants of attacks (McHugh et al., 2000). STAT IDS also does not have any wireless intrusion-detection components.

NAME	TYPE	Wireless Component?	Proactive?
SNORT	Public Domain	Yes	No
SHADOW	Public Domain	No	No
ISS RealSecure	Commercial	Yes	Yes
NFR Sentivist	Commercial	No	Yes
NetProwler	Commercial	No	No
Tripwire	Commercial / Public Domain	No	No
Cisco Secure	Commercial	No	Yes
Emerald	Research	No	No
STAT	Research	No	No

Table 4.2: Summary of IDS's properties

Above in Table 4.2 all the IDSs described in this section have been summarized allowing one to get an overview of all the IDSs compared with one another.

As can be seen, there are many IDSs available at the moment. Some of these IDS products are research based and others public domain or commercial IDSs. One can also see from Table 4.2 that within most IDSs with the exception of ISS RealSecure, there is a real need for the IDS to be updated with both wireless and wired intrusion-detection capabilities. Many of the IDSs listed, also lack ability to proactively detect attacks and limit or stop the attack before it can release its full payload. Two questions that should be asked are how effective are these IDSs in accurately detecting intrusion attacks and what limitations do currently available IDSs have?

4.5 Limitations of Current Intrusion Detection Systems

In the previous section it was clearly indicated that most of the commercial IDSs do not have a wireless component to detect or prevent wireless attacks. However, together with the absence or poor operation of “wireless” IDSs, research also claims that IDSs have many other limitations, and one has to understand all the limitations before defining an improved IDS (Kemmerer & Vigna, 2002). Therefore, these limitations need to be addressed so that the future IDSs might have even better results with far fewer attacks slipping through due to defects / limitations. This section highlights the limitations that must be taken into consideration when defining a new IDS, with the main focus on wireless protection.

Some of the activities that IDSs perform, and the associated benefits that they provide an organization, are first discussed. One needs this background to understand the limitations and flaws that IDSs today have associated with them. These flaws are discussed later in the section, but first one needs to know what an IDS can and cannot do when performing its functions. Below is a list of functions that current IDSs can and should perform according to Bace (1999).

1. Monitoring and analysis of both user and system activity;
2. Auditing of system vulnerabilities and configurations;
3. Assessing the integrity of critical system and data files;
4. Recognition of activity patterns of known attacks;
5. Statistical analysis of abnormal activities and patterns;
6. Operating system audit trail analysis and recognition of policy violations.

Although there are many benefits to having an IDS installed in the network, IDSs have some limitations that may leave the organization, that they should be protecting, open to attack. Below is a list containing some functions that an IDS **cannot** perform for an organization (Bace & Mell, 2001; Bace, 1999).

1. Compensate for missing or weak security mechanisms in a security infrastructure. This includes mechanisms such as firewalls, access control, authentication, link encryption and antivirus.
2. Instantaneous detection, reporting and response to an attack when there is a heavy load on the network or computer system resources.

3. Detection of newly published or variants / mutations of existing attacks.
4. Effectively respond to attacks perpetrated by sophisticated attackers.
5. Automatically investigate an attack without human intervention.
6. Resist attacks that are specifically designed to defeat or circumvent the IDS itself.
7. Compensate for problems with the fidelity of information sources.
8. Deal effectively with switched networks.
9. Cannot compensate for weaknesses in network protocols.

As can be seen by the aforementioned list, intrusion detection systems do have some limitations, most of which may be fixable through proper planning, maintenance and updates. This applies to both the IDS itself and the other security mechanisms within an organization's security arsenal. There are some limitations in wireless IDSs, which are unique to the wireless environment and do not affect wired networks. Three examples have been listed below (Airtight Networks, 2006). These are limitations that need to be taken into consideration when designing an IDS that is capable of detecting wireless attacks/intrusions.

1. **Neighbouring signals:** Within a WIDS it is difficult to differentiate between organizational and neighbouring organizations' wireless signals. This is not a problem within a wired environment IDS, but with a WIDS, neighbouring signals can lead to false alarm rates increasing. There is a second problem with this, in performing automatic detection and prevention because it is illegal to perform detection on a neighbour's signal. This can be a problem when determining which is the organization's signal and which is not.
2. **Location:** Wired IDSs can locate an intruder through tracing the attack back down the line and disabling the port on the switch or router to stop the attacker. With a WIDS this is much easier said than done: finding where an attack is originating from the airwaves is a daunting task.
3. **Security Planning:** In wired IDSs, the IDS can see all traffic on a line it is connected to and monitor it for attack. In wireless, this is a little harder to accomplish, as blind spots occur in the security realm because of lack of WIDS sensor coverage.

There are also many common defects that exist in intrusion detection systems, many of which may be due to poor programming practices and the likes. Below are listed some of the important defects (Lippmann et al., 2000; Wang & Knight, 2000).

Defect 1: Attack IDS Data Log

One of the first defects that today's IDSs have has to do with the way that IDSs log attack data. IDS vendors do not have a common file type or even logging method so that if an attack spans multiple IDSs, the attack data can be correlated. Take for instance a network worm, which attacks the network as a whole, thus triggering the NIDS to log the attack. At the same time, the worm corrupts files on individual host machines, thus triggering the HIDS on the machine to log attack data. Without a common standard to correlate IDS logs, the administrator will have a really tough job finding out where the attack came from or even if the HIDS and NIDS attack alerts point to the same attack (Becker & Petermann, 2005).

Defect 2: Inability to detect new attacks

The second flaw that modern IDSs have is the inability to detect new attacks. This can be seen in a 1998 study of six research sites, in which all data were analyzed after a seven-week set of training data and two weeks of test data were applied to the sites. The data contained 300 variants of 38 different attacks. These were embedded randomly throughout the training and test data. The reason for this study was to determine the attack detection rates of IDSs as a function of false alarm rates. The best systems had detection rates of 60% correct, when the false alarm rate was below ten false alarms per day for both old and new attacks, where a local user elevated himself to administrator (Lippmann et al., 2000). When it came to DoS attacks, the detection of older, already known attacks was above 80% correct. The problem came in with most systems in the detection of new, novel or mutated attacks, which had a lower than 25% correct detection rate, even with a lot of false positives (Lippmann et al., 2000). These figures show that IDSs need to be updated to enable them to more accurately detect new and novel attacks and not just the already known attacks. Even though this was an off-line study, the results of live tests correlate quite closely with the findings (Lippmann et al., 2000).

Defect 3: False User Training

Anomaly detection-based IDSs suffer from a defect that is very difficult to detect and prevent from happening. This flaw is that of when a user knows that an anomaly-based IDS is in use, and they wish to attack the system it is protecting, they can slowly, over a period of time, perform their attack (Wang & Knight, 2000). These slight deviations in the user's activity will usually go unnoticed and will allow the user to setup the system, so that when they are ready to attack, all the preparations will have been done, and the administrator may not even be aware that an attack has taken place (Wang & Knight, 2000). This is due to the IDS thinking that the user's actions are normal and not anomalous as they actually are.

Defect 4: System User Attack

Users operating at a low level, such as Windows system user, a level below which system auditing occurs, can actually thwart the auditing process on their attack and slip under the IDSs radar (Wang & Knight, 2000). This is a defect that is difficult to address, primarily because the problem sits under the actual IDS itself, and the IDS would need to have lower-level access to the system and monitor the low-level users differently. This is one of the easiest ways to circumvent an IDS. Thwarting the actual information collector of an IDS allows one to perform at least part of an attack without detection (Wang & Knight, 2000).

Defect 5: Inability to Cope with Networks

Many IDSs in existence today use detection methods that are four to six years old: technologies that were designed to run on networks running a maximum bandwidth of 100Mbit/s (McAfee, 2003). The problem with this is that networks have advanced to the point as described in a previous section of networks with 10Gbit/s capabilities. These IDSs may have difficulty detecting attacks on the faster networks and their detection engines may miss attacks and drop packets, as they are unable to keep up with the sheer volume of traffic (McAfee, 2003; Hutchinson, 2004).

Defect 6: High False Alarm Rates

One of the most worrying problems in current IDSs is that of high false alarm or false positive rates. IDS products available at the moment generate too many false positives and are lacking in both accuracy and specificity in their detection of attacks (specificity being how much and how detailed the information collected about an attack is) (McAfee, 2003; Hutchinson, 2004). False alarm rates are currently measured as a total number of incidents per day and not as a percentage of total actual alarms sounded (Lippmann et al., 2000). This does not show a true reflection of how well the system is working as a whole.

This section gives one a grounding in what an IDS can and cannot currently do. Many individuals think that by implementing an IDS, the security concerns of their organization will be addressed and that the network upon which it has been installed is now safe. This is simply not true. IDSs, as with any security software, have their limitations, which were discussed earlier in this section, including the limitations of the WIDSs, which are relatively new to the IDS market.

Currently IDSs have some defects that have been identified. While these defects should not deter one from implementing an IDS, if an organization knows the weaknesses beforehand, it can remember this information when planning the overall security solution. The next section details some of the characteristics which make a good WIDS. With this background, one can also see the limitations of current products.

4.6 Characteristics of Wireless Intrusion Detection Systems

As stated by Lim, wireless-based networks are difficult to secure due to the broadcast nature of wireless technologies and are open to both active attacks and passive intrusions (Lim et al., 2003). To combat these problems, wireless IDSs have some added features and characteristics that existing IDSs do not take into account. Most tools that perform security on networks operate at Layer 3 of the OSI model, explained in Chapter 3, and operate under the assumption that the lower layers of the network are secure (Lim et al., 2003). This is just not always the case

with a wireless network, and IDSs should also take this into account and operate at Layer 2 of the OSI model. This section outlines those features and characteristics that a wireless IDS should contain, based on operations performed at layers 2 and 3; as well as what should be considered when implementing an IDS on a network containing wireless technologies.

Currently, most wireless IDSs use a mixture of hardware and software known as Intrusion Detection Sensors for their implementation (Hutchinson, 2004). These sensors are placed in strategic positions on the networks and examine all traffic originating from, headed for and on the wireless network itself (Hutchinson, 2004). According to Salmanian et al. (2004), most attacks can be identified by attributes or identifiable signatures that distinguish them via attributes contained in the IEEE 802.11 MAC (Medium Access Control) and physical layer specification.

For example, a man-in-the-middle attack can be identified usually by looking at **Layer 2** packets flowing over the network and checking the packet against the signature database of known attacks. This allows the IDS to determine whether or not the packet is part of an attack. Since Wireless IDSs should monitor operations on **Layer 2** and **Layer 3**, aspects such as distributed detection, probe monitoring etc. should influence the characteristics of a wireless IDS. Based on some of the attacks identified by Lim et al. (2003), a wireless IDS should have one or more of the following characteristics in order to be effective.

4.6.1 Characteristics of Wireless IDS (WIDS)

- **Distributed Detection:** Because it is difficult to detect where an attack has originated from, as discussed earlier, WIDSs should contain distributed sensors. This is so that the WIDS can become more usable and able to more accurately determine where an attack originated from on the wireless network.
- **Detection of Rogue Access-Points:** This job is usually done manually by a system administrator, but WIDSs should have the capability to automatically detect and alert management of rogue access-points on the network. These rogue access-points are used by attackers to steal usernames and passwords.

- **Detection of Unauthorized Access-Points:** Sometimes within organizations, staff or visitors can plug their own access-point into the network by plugging it into an open RJ45 jack in the building. This provides a direct link into the organization's wired network as these unsanctioned APs most likely bypass WIDS sensors. WIDSs should have capabilities to detect these APs and alert the appropriate administrators.
- **MAC Address Blacklisting:** When an attacker has been spotted by either the IDS or administrator, the WIDS needs to add the attacker's MAC address to a table of blocked MAC addresses so the attacker will not be able to attack again using the same wireless device. A whitelist should also be kept of all organizational device MAC addresses, so they will not accidentally be blacklisted.
- **Probe Monitoring:** WIDSs should be able to monitor organizational probe requests to determine whether the client sending the probe request is actually allowed access to the AP or not. If an attacker is probing the APs, then he should not receive a probe response from the AP; this would have to be determined by the WIDS. Attackers often send out probe responses, flooding them over the airwaves in hopes that some client will connect to them and then they can attack the client (Hutchinson, 2004). WIDSs should also monitor for this kind of attack.
- **Flood / DOS Detection:** There are many forms of wireless flooding attacks, so the WIDS should have the capacity to detect and notify the system administrator that the wireless network may be under flood / signal jamming / DOS attack.
- **Access Point Failure Logging:** Most APs do not log all their errors, so the WIDS should monitor for events, such as authentication, association and dissociation. The WIDS should then generate logs and reports of these activities. The logs can then be later analyzed for signs of possible attack.

IDSs currently available on the market do not contain many of the above-listed characteristics that are needed to enable the IDS to detect and stop attacks from occurring. The model that is proposed in Chapter 5 takes these characteristics into consideration from

the inception of the model and provides a proactive generic model for both wired and wireless intrusion detection.

4.7 Conclusion

There are currently many IDSs available on the market, many of which come from very reputable security-focused companies. However, there are many defects and limitations associated with these IDSs. Proof of this statement can be found in the many new intrusion attacks reported yearly (Gordon et al., 2005).

All the background information about security, IDSs, their limitations and defects have been presented in this and previous chapters. From this background information it can be clearly seen that there is a need for a new IDS model: a model that not only addresses the problems and limitations associated with current IDSs, but one that takes into account the new problems associated with wireless networks.

Wireless networks, as stated previously, have their own limitations, problems and characteristics that do not allow a regular IDS to detect many of the attacks associated with them. IDSs are also currently lacking in stopping attacks before any serious damage is done, as most only notify an administrator of attack. As more attacks are constantly being developed and used, one can see that any new IDS model would need to be proactive in nature. Chapter 5 introduces NeGPAIM (Next Generation Pro-Active Identification Model) and its updates, which enable organizations to detect both wired and wireless attacks proactively.

Chapter 5

The Updated Proactive Identification Model (NeGPAIM-W)

5.1 Introduction

Over the past few decades, the Internet and networks have become a vital part of any organization's business processes, from selling products online to application sharing and distributed data processing. Hardware developers realize this and focus more and more on ways and means to improve networks and the Internet. One such development is that of wireless networks. The improvement in Internet and networks have not only improved business processes, but also increased the need to protect the business's information and access to information (DTI UK, 2004).

Each year, the number and complexity of attacks and intrusion incidents occurring against organizations is increasing, and although organizations feel safe, it has been shown that failures in security mechanisms are also increasing annually, e.g., in firewall or intrusion detection systems. This is a problem that was discussed in detail in Chapter 4 under the section *Limitations in Current Intrusion Detection Systems*. It was concluded that currently available IDS products do not adequately protect a system against new, mutant and previously unknown attacks, thereby leaving gaping holes in the security of organizations placing their trust in these IDSs.

This chapter focuses mainly on how to address these problems, discussing the shortcomings of the NeGPAIM Model (Botha, 2003) and thereafter, updating Botha's model. The Botha model was developed in 2003 and was based only on wired networks. To be in line with new developments, such as wireless networks, it is quite clear that the NeGPAIM Model needs some

updating. The updates allow the model to be applied to both wired and wireless environments, thus enabling the NeGPAIM IDS to detect attacks previously undetected.

5.2 Design Specifications for Updated Model

Over the past few years since the definition of the NeGPAIM Intrusion Detection Model, there has been a massive growth in the use of wireless technologies within organizations. This is mainly due to the benefits that wireless network technologies give to the organizations that use them. Although there are many benefits gained by using wireless technologies, such as mobility, increased productivity and ease of access, it should be noted that this all comes at a cost, as the wireless technologies are known for their inherent lack of security. These factors should all be taken into account when determining what changes need to be made within the previous model, allowing the new model to cater for the problems currently existing.

The current NeGPAIM Model has been thoroughly tested at the NMMU, and results received from the testing have been very good (Botha, 2003). Some of the results from testing include an identification rate of around 95 percent and a false detection rate of less than 5 percent (Botha, 2003). The NeGPAIM Model currently has little work done on wired networks and no work currently done towards the detection of wireless attacks. With results such as those described above, it can be seen that the model is very successful and only needs additional components that allow for the accurate detection of newer wired intrusions and complete detection of wireless-based intrusion attacks. These new components would be sensor based, due to the nature of networks, both wired and wireless, allowing the IDS to gain insight into all areas of the network via the sensors. For the reasons outlined above, the rest of this chapter focuses on the network-based data sources. For further information on the other parts of the model, please refer to Botha (2003).

The previous three chapters introduced intrusion detection along with its functional and non-functional components, including the Updated Generic Proactive Intrusion Taxonomy presented in Chapter 4. This focused on defining a generic process of understanding and identifying intrusion attacks. By utilizing the information gained through the literature studied, including

research done at the NMMU (Botha, 2003), it is now possible to define updated design specifications to Botha's NeGPAIM Model. The design specifications are illustrated in Figure 5.1.

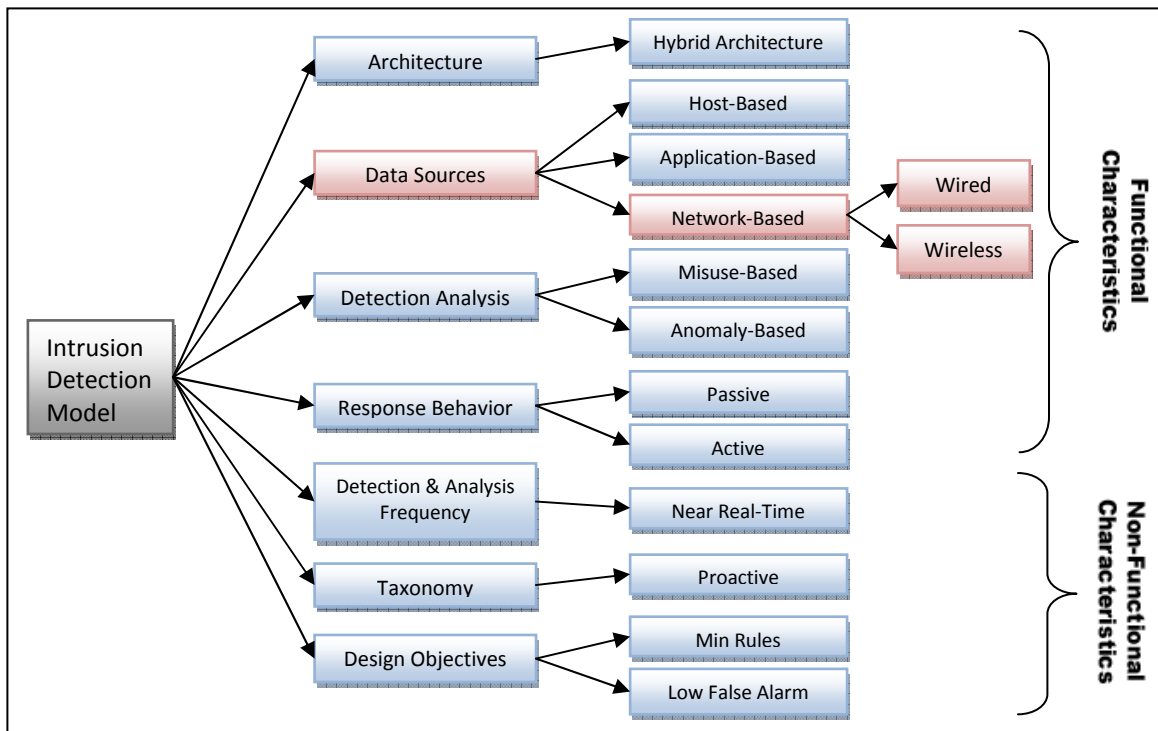


Figure 5.1: Updated Model Design Specifications

Figure 5.1 depicts the logical splitting of the Intrusion Detection Model into two distinct sections: **Functional Characteristics** and **Non-Functional Characteristics**. The primary elements of Functional Characteristics are: Architecture, Data Sources, Detection Analysis and Response Behaviour. The IDS model incorporates two distinct detection methodologies, namely Network-Based Detection and Host-Based Detection together forming a hybrid IDS methodology (Botha, 2003). This hybrid methodology allows the model to take data from host, network and application based sensors as input. The analysis of attack/intrusion data is performed by utilizing both misuse and anomaly detection methodologies, allowing for a more holistic attack determination (Botha, 2003). If the model detects an attack, it contains response behaviours that can implement both passive and active response to the attack/intrusion.

Certain components of the design specification have been updated with regards to the data sources, specifically the network-based sensors. The network-based data source has been updated to include both wired and wireless-based network sensors. This allows the misuse and anomaly detection engines access to more updated network information, including the physical media type the information was collected on. The active and passive responses also contain distinctions between attacks originating on wired and wireless networks, as there are many differences between the mediums.

The Non-Functional Characteristics' primary components consist of the following: Detection and Analysis Frequency, Taxonomy and Design Objectives (Botha, 2003). The model allows for the collection of input data from sensors, as well as the processing of this data through the two detection engines in near real-time speeds. Through the use of the updated generic proactive taxonomy, the model allows the IDS implementing it to detect intrusive behaviour and allow the administrator to follow the intruder during his attack/intrusion. The design of this model allows for the IDS implementing it to contain fewer detection rules than other IDS models, because of the nature of the detection engines and their relationship.

The seven main characteristics mentioned above, both functional and non-functional, were implemented in the model. If the updates to the existing characteristics are successfully integrated into the updated model, the model's main objectives have been met; thus, allowing for an IDS, with a high rate of attack detection, including known and unknown attacks, sourced from both wired and wireless network segments. The updated model's effectiveness in this respect are in part determined by the hybrid architecture. The hybrid architecture is discussed in the next section.

5.3 Hierarchical Hybrid Architecture

The previous section explained that hybrid architecture is needed and plays an important role in determining the overall effectiveness of the updated model. Questions about how and where both the network-based and host-based components have been implemented are best defined by the structure of the hybrid architecture. The hybrid architecture is based on three layers: namely

input, process and output layers. These layers are known as the System Model, used to represent many ideas in the IT and computer-science fields (Armstrong, 2001). The model is best defined by utilizing the system model's hierarchical structure. The focus of this section is mainly on the network-based IDS components, which will be described in terms of the system model layers. Each of the layers is described below and is represented graphically in Figure 5.2.

- **Input layer:**

The intrusion detection system gains all its input data both from host-based and network-based sources, via the input layer. The input layer consists of multiple sensors, which can be either host-based or network-based. A network-based sensor could be, in turn, either a wireless network sensor or a wired network sensor. An example would be a sensor "A" placed near a wireless access-point, enabling the IDS to gain information on traffic on the particular wireless network segment from sensor "A".

The same is true for a wired network sensor. The sensor "B" would be placed in a section of the wired network where it could gather information on the traffic flowing through it. This is the main difference between wired and wireless sensors: wired sensors get placed in sections of network where all traffic has to pass through them, whereas wireless sensors have to actually sniff the wireless traffic. The wireless and wired sensors differ slightly in the way they forward the information to the processing layer. The reason for this is the slight latency that can occur on a wireless network. The timing of the wireless sensors have to take this latency into account. The updated model also allows for data to be captured at both layers 2 and 3 of the OSI model, whereas the previous model focused on the capture of, primarily, layer 3 data. The capture of additional layer 2 data allows for more holistic input.

Host-based sensors, on the other hand, gather information for the IDS from the operating systems audit logs and audit logs of the domain / directory service server. Host-based sensors can also take the form of application-based sensors, where information is gathered from various application programs running on a host.

- **Processing layer:**

The processing layer is where all the input data collected by the various host and network-based sensors is processed for signs of intrusion. The processing is also taken into account whether the sensor was a wireless or wired network sensor when determining the type of processing to be done and which intrusion/attack definitions to use. The processing layer consists of three components split into two processing layers seen in Figure 5.2.

The first layer is the low-level processing layer and consists of two processors: one implementing the misuse detection approach and the other implementing the anomaly detection approach. The previous model, while quite complete in its host-based detection, lacks a holistic approach to network detection as wireless networks were not taken into account at the time. The accuracy of the model is, thus, slightly off as it would be unknown how many wireless attacks would be missed by the model. The updated model improves the overall accuracy by taking wireless attacks into consideration. This, coupled with the updates to the wired detection rules, allows the new model a far greater detection accuracy. The two low-level processors have gained increased detection accuracy, also in part from the input layer's updates allowing layer 2 and layer 3 data collection, which allows the low-level processors access to more information when making attack determination.

The low-level processors have been updated to work on a weighted system, allowing a weighting to be set for attack data sourced on the wireless network, and a separate weighting set to attack data sourced on the wired network. These two weightings allow the engines to function more accurately and also enable their output to be a better reflection of the actual attack being perpetrated as a whole.

The two low-level processors feed their outputs to the high-level processor that acts as a central analyzer, determining overall attack status by combining the outputs of the two low-level components. The output of the central analyzer is forwarded to the output layer.

- **Output layer:**

The output layer takes the information it gains via the processing layer and determines what response would be appropriate for the event. This response could be either passive or active, depending on the severity of the intrusion attack. The responses could be configured according to the needs of the organization implementing it. Depending on the type of attack being perpetrated, responses could be aimed at stopping the attack at either layer 2 or layer 3 of the OSI model, allowing for the best chance of mitigating the attack.

The administrator has a warning of attack set in the management console. Included in this alert will be the individual engine's outputs with its attack weightings. The severity of the attack is also listed, and the administrator is advised on actions to take. These actions could be hardware, software or legal actions.

While this section described the updated model in terms of its input, processing and output, utilizing the system model, the next section proposes updates to a conceptual model developed at the NMMU.

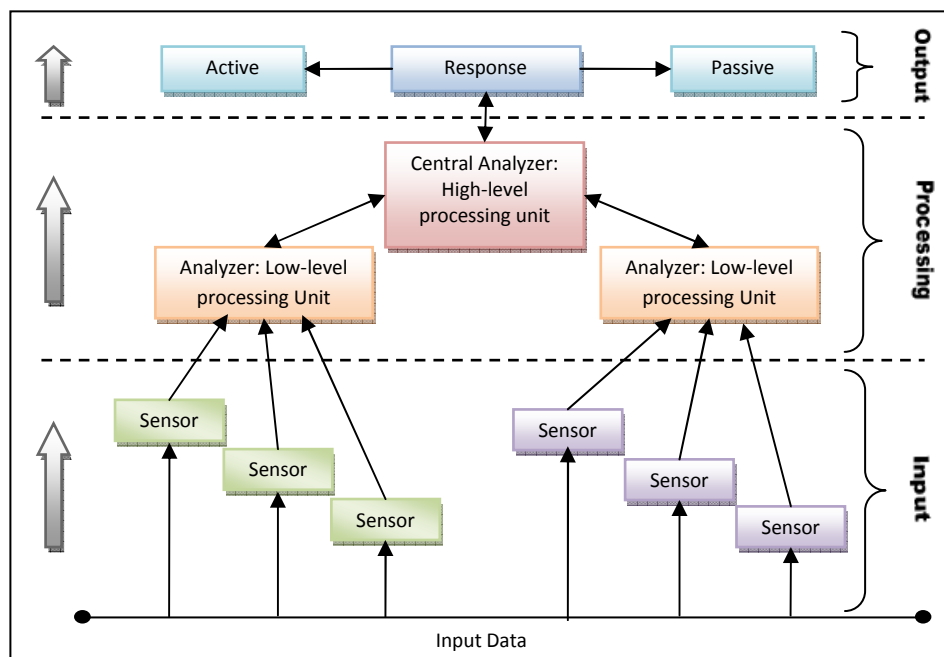


Figure 5.2: Hierarchical Architecture (Botha, 2003)

5.4 The Conceptual Model: The Components

This section proposes updates to the model developed at the NMMU, which is possible due to the design specifications and the hierarchical hybrid architecture introduced in the previous sections. The model to be updated is known as the Next Generation Proactive Identification Model (NeGPAIM). The updated model is called Next Generation Proactive Identification Wireless Model (NeGPAIM-W).

The model is dependent on nine major components. These core components are known as: Information Provider, Collector, Coupler, Information Refiner, Fuzzy Engine, Neural Engine, Central Analysis Engine, Responder and Manager. These components can be seen in the graphical representation of the model in Figure 5.3 below.

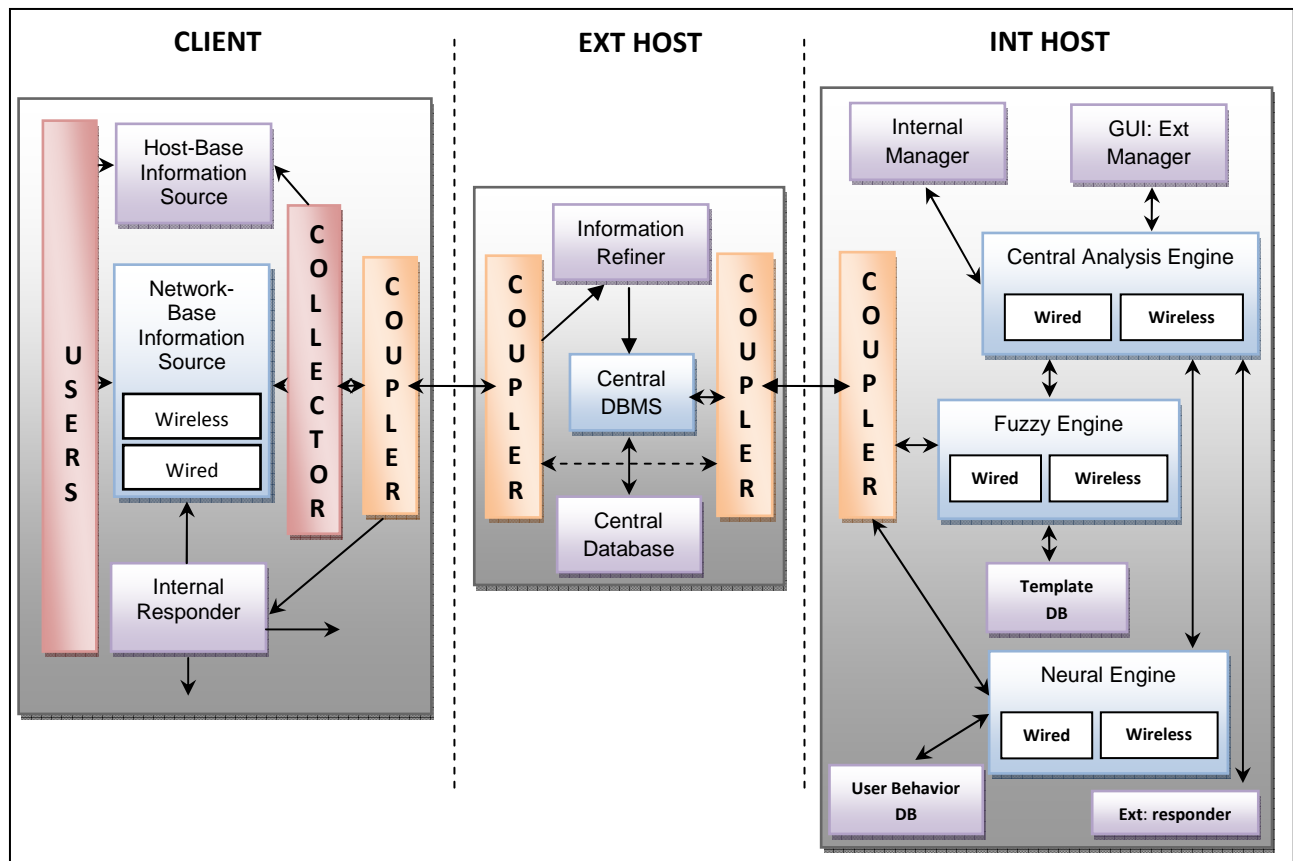


Figure 5.3: General Representation of NeGPAIM-W

The major components of the updated model have been discussed below, these components are divided into a three-tier hierarchical hybrid architecture as seen in Figure 5.3. The three tiers are as follows: Client, External Host and Internal Host. This architecture allows for benefits of performance and security to the NeGPAIM-W Model and allows the security administrator the ability to monitor his/her network for attack more efficiently.

Each of the components that make up the updated NeGPAIM Model is briefly discussed below.

- **Information Sources:**

Refers to the different providers of input data into the NeGPAIM-W Intrusion Detection System, which include host-based, network-based and wireless information sources. This paper focuses on the wireless information sources.

- **Wireless Network-Based Information Source:**

This information provider collects information about the user's activities on the wireless LAN, allowing the IDS to gather statistics on usage and the user's activities within the wireless environment. The information is gathered primarily from the access points and servers containing IDS sensors, which are located around the organization's offices.

- **Wired Network-Based Information Source:**

This information provider works by having sensors sitting on network servers and on devices around the network. They monitor for network-based attacks, such as DoS, DDoS and spoofing attacks and report back on the network as a whole.

- **Host-Based Information Source:**

This information source collects and returns information collected from a specific host on the network, and it is only concerned about attacks aimed at the host itself or attacks emanating from the monitored host.

- **Collector:**

This component is a Windows service that has the responsibility of collecting information from the information sources and forwarding the data to the information refiner.

- **Coupler:**

The coupler is an interface that allows for the three tiers, namely: client, external host and internal host to interact and share information.

- **Information Refiner:**

This is responsible for converting data into a format usable by both the fuzzy and neural engines.

- **Fuzzy Engine:**

The fuzzy engine is one of the two low-level processing units of NeGPAIM-W, the second low-level processing unit is the Neural Engine discussed next. The low-level processing units differ from the high-level unit by the fact that they do the processing of raw attack input data. This engine is responsible for implementing the Misuse Detection methodology and computes a template firstly so that the user action graph is mapped against it to determine whether or not a user (intruder) has been or is performing an intrusion attack.

The overall intrusion probability for the network sensors is divided into two weighted parts: one weighting for the wireless attack probability and another weighting for the wired network intrusion attack probability. The fuzzy engine's network detection rules have been updated with the new NeGPAIM-W Model to provide a better detection rate. The rules have been updated to detect attacks at layers 2 and 3 of the OSI model where the previous fuzzy engine specifically targeted layer 3 only. These updated rules detect intrusions by the use of sensors detecting layer 2 attacks on the source network medium and layer 3 attacks on servers and workstations. The updated model allows for fast detection performance by separating the network detection into wired and wireless separately, and weighting the outputs to form a final fuzzy intrusion attack probability. This also allows the engine to take into account the differences in transmission of data over the different network mediums.

The fuzzy engine passes its intrusion probability value to the central analysis engine. This is a continuous process.

- **Neural Engine:**

The neural engine is the second of the two low-level processing units and also processes input data. This engine processes the data and searches through it for patterns of abnormal user behaviour that may be occurring.

This abnormal user behaviour may come in through one of three sources: the host-based sensor, the application-based sensor or the network sensor. The network-based sensor is what this section explains. The neural engine uses a user's wireless and wired network usage patterns to determine whether or not the user is acting abnormally on the system. For instance, the user may work via a wired terminal from 8am to 5pm, Monday to Friday. Then one day he/she logs into the network on a Sunday afternoon over a wireless connection. This is noticed by the neural engine as anomalous activity.

The engine reports abnormal user behaviour to the central analysis engine by way of intrusion probability value. This intrusion probability value or IPV is the output of both the Fuzzy and Neural engine and is a percentage probability of attack determined by the engine.

- **Central Analysis Engine (CAE):**

This is a high-level processing unit, the objective of which is not to perform anomaly or misuse detection, but rather to analyze and interpret the resultant output values from the fuzzy and neural engines as well as managing the other units of the model.

The inputs received from the fuzzy and neural engines have their probabilities put through in weighted form for the probability of network attack, and these weightings are used in the determination of what kind of attack is taking place, and where the source of the attack is. This enables the correct responses to be applied to the attack, e.g., a predominantly wireless attack has the wireless part of the

attack patched first, as that would be the main part of the attack and without it, the attack may cease.

The engine outputs a final intrusion probability with final weighted scores of attack type, source, etc. This is generated after performing statistical calculations on the output of the two lower-level units.

- **Responder:**

The responder is responsible for taking the necessary action in the event of an intrusion attack. The responder can either respond via passive or active responses. Active responses in the case of wireless intrusions would be to block the intruder's MAC address on the wireless LAN. A passive response would be to alert the administrator to the possible intrusion via e-mail.

- **Manager:**

This component allows for the management and configuration of the intrusion detection system. The manager also allows the administrator the ability to see what attacks have occurred and the means to access the responder, effectively stopping an intrusion attack.

The next section helps one to better understand the changes made in the NeGPAIM-W Model. These changes also help one to better understand how the updated model identifies wireless based attacks.

5.5 The Model in Perspective

The previous section identified the nine components making up the NeGPAIM-W Model. This section identifies some of the differences making the updated NeGPAIM-W Model better able to identify wireless attacks. This allows one to see how the updated model is better than the current model at overall intrusion attack detection and prevention. Components that have been updated are focused on in more detail.

The functional characteristics of the data source identified in Section 5.2 are implemented via the following components of the model: Information Provider, Collector and Information Refiner. The information provider within the updated model has been updated primarily with regards to its network-based information source. The changes to this information source, by the addition of wireless network detection, allow the model to detect a wide range of network attacks. These attacks would have previously gone undetected due to the current model's inability to detect wireless specific intrusion attacks.

The three primary components of the NeGPAIM-W Model are the Fuzzy Engine, Neural Engine and the Central Analysis Engine. These components are responsible for the actual detection of intrusive behavior and misuse from the information gathered by network-based and host-based sensors. The three engines are the implementation of the detection analysis functional characteristic of the design specification from Section 5.2. The fuzzy engine differs slightly from the current fuzzy engine models as it has been updated to include detection rules for the newly added wireless network updates. The fuzzy engine's output to the central analysis engine has been updated as well, so that the output is now distinguish between attacks detected on the wireless and wired LAN to allow for the appropriate weightings assigned to the attack.

These changes have also been implemented to allow the responder to determine which, if any response, is to be fired. The responder is also be aware of which network segment to apply the responses to. The responder implements the response characteristic of Section 5.2's design specification.

This section has shown the model's updates as they relate to the design specification and has also shown how the updated model differs from the current NeGPAIM Model. The next chapter discusses the processing components of this model in more detail. Again, the focus is on the wireless components.

5.6 Conclusion

Intrusion attacks are growing in numbers each year and current IDSs are generally unable to detect attacks proactively and respond to them quickly. This is why the NeGPAIM-W Model is indeed needed, as was concluded in Chapter 4. The model's objectives are to allow for the proactive identification of intrusion attacks on wireless, wired networks and on the host itself. It is envisaged that the last mentioned is done with a high rate of detection and low number of false alarms.

This model, if implemented correctly, should give any organization's security officer or systems administrator the ability to detect attacks on the network holistically, as opposed to only one network medium. It also provides the organization with a means to fight back at the ever-increasing number of attacks coming out each year.

The next chapter focuses on the processing components of the model, and focuses on the fuzzy logic, neural networks and statistical calculations that make the engines function. These engines form the heart of the model and can be used to combat intrusion attacks proactively and accurately.

Chapter 6

The Fuzzy, Neural and Central Analysis Engines

6.1 Introduction

The Next Generation Proactive Identification Wireless Model (NeGPAIM-W) was introduced in the previous chapter. This model, as was discussed previously, can proactively identify and protect a system from intrusion attack, be it known or previously unknown to the model. This model is more effective in overall network protection than its predecessor, due to its wireless detection components.

This chapter focuses on the three engines that enable the NeGPAIM-W Model to detect attacks in a proactive manner. The three engines are namely the fuzzy, neural and central analysis engines, each adding to the overall detection mechanics. The first section of this chapter takes an in-depth look into the fuzzy engine, with the main focus on the reason for the choice of this detection mechanism and a detailed description of the detection methodology. Thereafter, the neural engine is discussed with the main focus again placed on the detection mechanism and detection methodology. This is followed by a detailed explanation of the central analysis engine, including a detailed description of the methodology and its role in the overall detection process. Finally, an example is given, allowing one to gain an understanding of the detection process from start to finish. The main purpose of the example is to highlight how the engines function.

6.2 The Fuzzy Engine

The fuzzy engine is the engine that has undergone the most changes in order for the implementation of wireless intrusion attack detection components. This section aims to describe

the fuzzy engine and the method of detection used to implement the engine. The main focus is on the additions to the engine allowing for the wireless component to the model. The purpose of this section is to understand the internal workings and the reasons for the existence of the fuzzy engine in wireless environments. The fuzzy engine has been explained in terms of the following:

- An alternative approach to misuse detection;
- Fuzzy methodology;
- The mapping strategy; and the
- Dynamic proactive identification model.

These four points form the basis for this section and enable one to better understand how the fuzzy engine functions.

6.2.1 Alternative approach to misuse detection

As stated previously in Section 2.5.1, misuse detection is utilized by the majority of IDSs (Bace & Mell, 2001). When looking at the problems associated with misuse detection, it is clear that there is a need to revise the misuse detection approach. The main objective must be to allow for a more dynamic method of performing misuse detection than is currently available. This section introduces, define and focus on updating the IDS dynamically for both wireless and wired network attacks.

Such an alternative approach to misuse detection must allow the IDS to accurately detect both known and new intrusion attacks in a generic fashion. In order to achieve this, the alternative misuse approach should focus on more accurate data as to the type and source of the intrusion, as well as the modus operandi of the attacker.

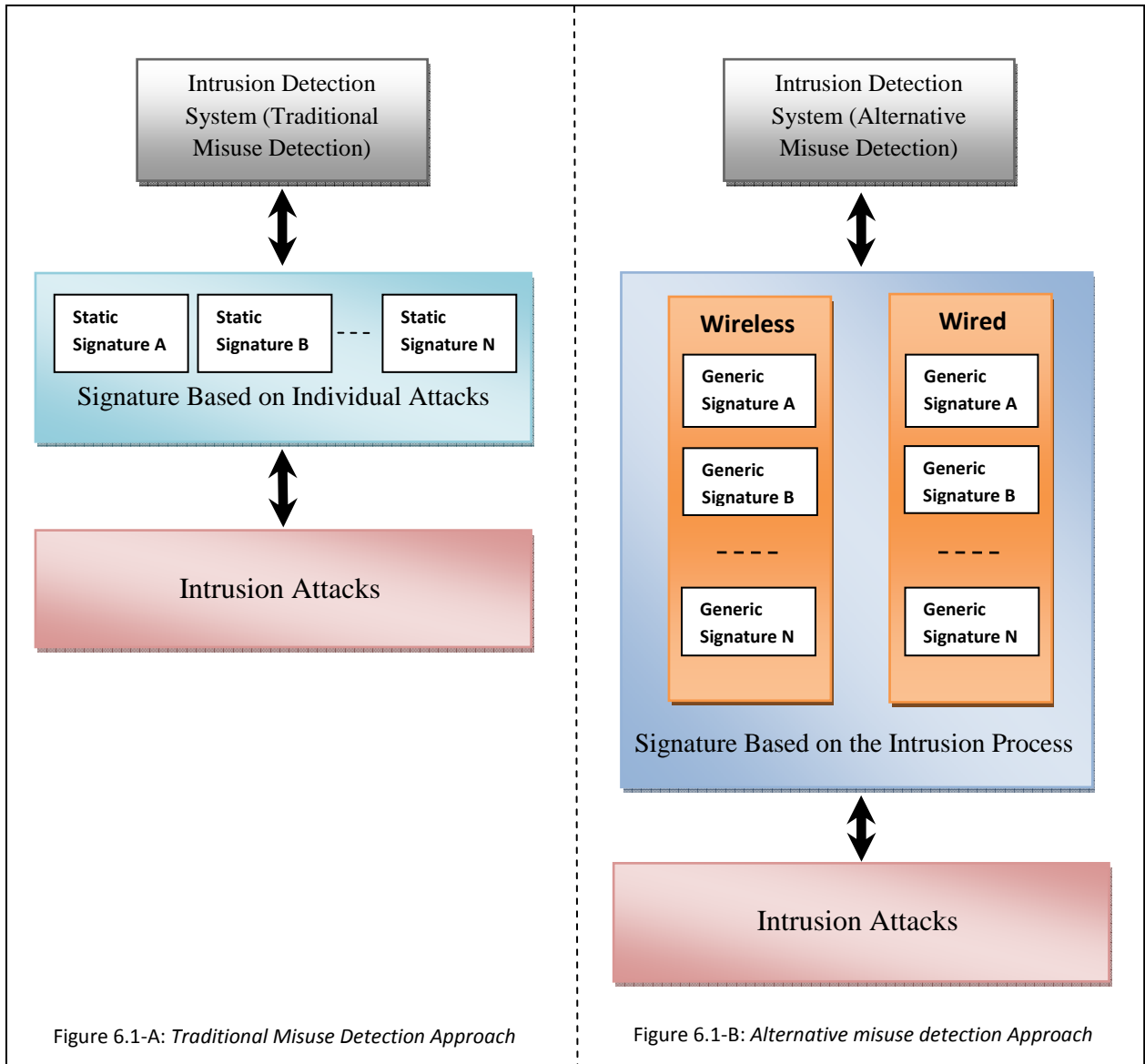


Figure 6.1: Comparison between Traditional and Alternative Misuse Detection

Figure 6.1 shows the differences between the traditional misuse detection approach seen in Figure 6.1-A, and the updated alternative misuse detection approach seen in Figure 6.1-B. With new signatures required to detect new attacks, the IDS must be updated regularly to allow it to detect attacks not in its signature database. Aside from the regular updates, another problem with this traditional approach is that the rate at which the IDS utilizes system resources as it scans through all known attack signatures to find a possible match to the attack taking place. The problem has a lot to do with the amount of data stored about each attack, as certain attacks have large payloads. Scanning through the vast amount of

information stored about each attack may take quite some time. Much of this information is unnecessarily stored, and many attacks contain the same data with slight mutations. Improving efficiency means that the focus of the alternative approach should shift from defining new signatures and regularly updating the IDS, as is the case in current misuse methodologies, and focus rather on generic intrusion signatures as seen in Figure 6.1-B.

Generic signatures share information amongst themselves, thereby drastically reducing the amount of information stored on each intrusion attack. This sharing of information occurs as follows: each generic sensor is assigned properties, for example, a generic DoS attack signature has properties assigned to it, such as a port range, bandwidth utilization, incoming packet size and network type. These are some of the properties that make up a DoS attack, although some of the properties mentioned may be associated with other attacks, e.g., the port-range property may also be a property in a port-scan attack. So as can be seen with the sharing of information by generic sensors, the information stored about attacks is cut down dramatically.

By the use of generic signatures, an IDS is more efficient in detecting both known and unknown intrusion attacks. These generic network intrusion signatures are further split into wireless network generic signatures and wired network generic sensors. The alternative misuse detection approach determines the source of attack data, e.g., whether the attack is a wireless-based intrusion attack or wired network attack. This allows for speedier attack detection by limiting searches through the signature database to signatures that correspond to the source network type. An example could be that an attack is occurring over a wireless network link. The attack database may have 100,000 network-attack signatures, 25,000 of which are wireless-attack definitions. With the network type having been determined as wireless, the search is effectively only a quarter of what it would have been with the usual misuse detection approach.

In IDSs implementing the current misuse detection approach, there is a lack in correlation between attacks that have been detected. This leaves the IDS without the knowledge of the bigger picture in terms of the attack, as some attacks serve as forerunners to the larger attack and ultimate payload. Any new attacks that the IDS does not have the signatures to might

also go unnoticed, and if it is part of a larger attack, it will most likely make the larger attack look not as deadly, because of the lack of knowledge of the whole attack. This problem has been addressed by rather than merely searching for intrusions by intrusion signatures, the new approach will search for intrusions, based on the logical process that intruders tend to follow when performing an attack against an organization's systems.

The resultant output of the engine implementing the alternative approach to misuse detection will output not just one output, as is the case of the traditional approach, but rather has two outputs. The first output will be the same as the output from the traditional approach, namely whether or not a user or intruder has, in fact, performed an actual intrusion attack. The second output is where the new approach takes the lead. The output at this stage will indicate the possibility that the intruder is still in the process of performing an attack. This will also contain the probability that the attacker is performing a purely wireless attack, a purely wired network attack or a mixture of the two to attain his goal. This will be stated through an intrusion probability value (IPV). The two-part result is calculated using generic signatures and a form of intelligent algorithm. This is done as the intelligent algorithm reads in the information gained by the various generic signatures and outputs an IPV of the attack. The IPV representing the percentage certainty that the actions performed represent an attack on the network. Thus each step completed by the attacker towards his/her goal e.g. probing open ports; brute forcing passwords etc. increases the IPV value. With the generic signatures detecting intrusion attacks based on the intrusion process as opposed to a specific event, the generic signatures only need updating if the way in which attacks are performed changes.

After researching the practical implementation of the alternative approach, some major shortcomings were identified. The first is the lack of precise and accurate data and the second is the lack of intelligent algorithms to identify intrusions. The rest of this section explains the shortcomings in detail, including how they may be overcome by focusing primarily on the networking aspects of the problems.

6.2.1.1 Identification of suitable precise data

This subsection explains how the lack of suitable precise data can be overcome. In this section, precise data refers to data that can be used accurately to determine an intruder's

actions. These actions could be illegal access to network resources, illegal access to application programs, etc.

This kind of precise data is usually very scarce, as most activity on a network or host is legitimate activity. So when data which indicates illegitimate activity is detected on the network, host or system, it must be used optimally. As was described in Chapter 4, sensors are one way that the IDS can gain this precise information on a system. These sensors can be wireless network-based, wired network-based or host-based.

The six generic intrusion phases identified in Chapter 4 are used in the alternative misuse approach. The six generic intrusion phases, which have been listed below, are now explained in terms of the alternative approach:

1. Probing phase;
2. Initial access phase;
3. Super-user access phase;
4. Hacking phase;
5. Covering phase; and
6. Backdoor phase.

Each of the six generic phases have many generic signatures associated with it. The reason for this is that when having many generic signatures, the system has more accurate information on the attack. Starting with the probing phase's signatures and continuing through the signatures within each phase, determination of whether the attack is emanating from the wireless or wired network will be made. The intruder is followed closely through the six generic phases as he/she progresses on the network. Each phase will follow the intruder's network/host usage, thereby monitoring the intruder throughout his attack. The purpose of the signatures is to track the intruder's/user's movements on the network/host throughout the six generic phases, as well as tracking his/her other actions.

The availability of precise and accurate data has been identified, as has been the relevance of the updated alternative approach to misuse detection. The next section describes and explains the algorithm that implements the updated alternative approach described previously.

6.2.1.2 An intelligent algorithm

As mentioned earlier, there are two main problems associated with the alternative approach to misuse detection. The previous section iterated the need to gather precise data. This section is dedicated to the second problem: the lack of intelligent algorithms to implement the alternative misuse detection approach.

Current algorithms implemented by intrusion detection systems have difficulty correlating and combining collected precise data with other non-precise data collected on the system (Valeur et al., 2004). This problem limits the IDS from looking at the attack as a whole. Therefore, the need for an intelligent algorithm is great, and if implemented, it allows the IDS to use all available data to track the intruder as he/she moves through the six generic intrusion phases. In order for the intelligent algorithm to do its job, it needs to perform the following two tasks:

- Combine all the data collected from various sources; and
- Interpret the combined data according to a transfer function, allowing it to determine the intrusion probability value.

There are many transfer functions available. Some common transfer functions include linear, non-linear, sigmoid and Gaussian functions. Due to the lack of accurate historical intrusion data, it is impossible to determine which function would be more effective in terms of the updated proactive generic intrusion taxonomy.

Thus, it was decided to define a new non-linear function, which is mapped directly to the proactive generic intrusion taxonomy and is based on a weighting structure. The weight structure is determined by the importance of each phase in terms of the four general intrusion result classes (namely, corruption of information, disclosure of information,

theft of service and denial-of-service) as outlined in the generic proactive intrusion taxonomy seen in Figure 6.1 (Botha, 2003).

A nonlinear approach was also used for implementing the fuzzy engine for the previous NeGPAIM model. This non-linear approach has been updated to allow it to be applied to the updated proactive generic intrusion taxonomy. Certain weightings would not have been accurate when applying the previous weighting scheme to wireless-based intrusion attacks. This can be seen by the importance of the phases identified between an attack taking place on a wireless network, as compared to a wired network attack. In a wireless attack, more attention should be paid to the probing and initial access phases, focusing on probe requests (Interlink Networks, 2002).

These phases in a wireless attack may take the intruder weeks to complete, as opposed to a wired attack, where the same phases may take hours or even minutes. An example captured packet of one such probing phase attack (unauthorized client connection) has been listed below in Figure 6.2. This figure shows a probe request of an unauthorized user/intruder actively scanning for an access point as one of the first parts to a wireless attack: locating a target. Once the target has been found, he/she will enter the Service Set Identifier (SSID) and attempt to connect (Interlink Networks, 2002). One can see from the source address field in the figure, that this is where the system will first gain knowledge of the intruder's Media Access Control (MAC) address from the source address field.

```

IEEE 802.11
  Type/Subtype: Probe Request (4)
  Frame Control: 0x0040
    Version: 0
    Type: Management frame (0)
    Subtype: 4
    Flags: 0x0
      DS status: Not leaving DS or network is operating in AD-HOC
      mode
        (To DS: 0 From DS: 0) (0x00)
        ....0.. = Fragments: No fragments
        ....0... = Retry: Frame is not being retransmitted
        ...0 .... = PWR MGT: STA will stay up
        ..0. .... = More Data: No data buffered
        .0.. .... = WEP flag: WEP is disabled
        0... .... = Order flag: Not strictly ordered
      Duration: 0
      Destination address: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)
      Source address: 00:02:2d:1b:51:ca (Agere_1b:51:ca)
      BSS Id: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)
      Fragment number: 0
      Sequence number: 1
IEEE 802.11 wireless LAN management frame
  Tagged parameters (13 bytes)
    Tag Number: 0 (SSID parameter set)
    Tag length: 9
    Tag interpretation: roguehost
    Tag Number: 1 (Supported Rates)
    Tag length: 4
    Tag interpretation: Supported rates: 1.0 2.0 5.5 11.0 [Mbit/sec]
0000 40 00 00 00 ff ff ff ff 00 02 2d 1b 51 ca @.....-.Q.
0010 ff ff ff ff 10 00 00 09 72 6f 67 75 65 68 .....rogueh
0020 6f 73 74 01 04 02 04 0b 16 ost.....

```

Figure 6.2: *Unauthorized Client Probe Request (Interlink Networks, 2002)*

One can see that the weighting structure needs to be updated, allowing more emphasis to be placed on the first two phases in a wireless attack. These initial phases are critical in the effectiveness of the attack as a whole and give the IDS information as to the intruder's modus operandi.

PHASE	ATTACK 1 (CORRUPTION OF INFORMATION)	ATTACK 2 (THEFT OF INFORMATION)	ATTACK 3 (THEFT OF SERVICE)	ATTACK 4 (DENIAL OF SERVICE)	WEIGHT STRUCTURE
1.Probing phase	3	3	3	1	10%
2.Initial access phase	2	2	2	1	20%
3.Super-user access phase	2	2	1	3	20%
4.Hacking phase	1	1	1	3	30%
5.Covering phase	3	3	2	3	10%
6.Backdoor phase	3	3	1	3	10%

Table 6.1: *Weighting Structure for Wired Network Non-Linear Function (Botha, 2003)*

Table 6.1 and 6.2 both show the six generic phases, and how they relate to the four types of attack objectives that an attacker has when attacking wired and wireless networks respectively. These four categories are as follows (Botha, 2003):

1. Corruption of information;
2. Theft of information;
3. Theft of service; and
4. Denial of service.

There are many attacks that fit into each of these categories and, as such, the explanation of the above table is described generically. Three levels of importance have been utilized to describe the significance of each of the six phases to the intruder in ensuring that he or she is able to achieve each class of intrusion result. Level one indicates that the level is the most important to the intruder, and level three indicates that the level is of less importance to the intruder. The weighting structure is determined by the importance of each phase in terms of the four general intrusion classes namely: corruption of information, theft of information, theft of service and denial of service. As can be seen

from Table 6.1, for an intruder attempting to corrupt information over the wired LAN, the intruder will more than likely have to perform the hacking phase, as this is very important for the attack of this nature.

This means that the intruder or hacker can skip any of the six phases, except the hacking phase, in order for his attack to be successful. The probing, initial access, covering and backdoor phases are not all that important to the hacker in achieving corruption of information. The weighting structure works well in determination of attacks performed against a wired network, but when applying the same weighting structure against a wireless attack, it will not produce the desired results. Therefore, an updated weighting structure has been devised and is seen in Table 6.2 below.

PHASE	ATTACK 1 (CORRUPTION OF INFORMATION)	ATTACK 2 (THEFT OF INFORMATION)	ATTACK 3 (THEFT OF SERVICE)	ATTACK 4 (DENIAL OF SERVICE)	WEIGHT STRUCTURE
1.Probing phase	1	1	2	1	30%
2.Initial access phase	1	2	2	1	20%
3.Super-user access phase	2	3	2	3	10%
4.Hacking phase	2	2	1	2	25%
5.Covering phase	3	3	3	3	5%
6.Backdoor phase	1	2	3	3	10%

Table 6.2: *Weighting Structure for Wireless Network Non-Linear Function*

As can be seen from the weighting structure in Table 6.2, the weightings on a wireless network's non-linear function are far different than those on the table of wired network weightings. The main differences can be seen in the probing phase, as users cannot simply connect to wireless networks: they first need the SSID, as mentioned earlier in this section. This means that attackers have to use a sniffer application, or put their wireless

card in monitor mode (mode allowing for promiscuous monitoring of packets) to find wireless networks (Interlink Networks, 2002).

Intruders performing wireless intrusions take a lot more care in performing the probing phase. This is to gain as much information about the wireless network as possible. Wireless intrusions are also harder to trace because of the lack of a physical cable, so the covering phase is far less important to a wireless intruder (Aruba Networks, 2004). Taking the above into consideration, the weighting structure on the wireless network linear function should be assigned to reflect the way wireless attacks occur.

The differences between the new wireless weighting structure and the older, but still relevant wired network weighting structure, can be seen in Figure 6.3 and Figure 6.4, explained later in this section. These differences can be seen if one looks at the importance of each of the generic phases for the denial of service (DoS) attack in Table 6.2. A wireless-based DoS attack, such as signal jamming, relies heavily on both the probing and initial access phase to gain information on the system and to gain an initial foothold on the network. In this attack, the covering and backdoor phases are not all that important to the hacker, primarily because it is extremely difficult to detect and catch an intruder hacking over wireless.

The weightings seen in Table 6.2 were settled upon after multiple wireless attacks had been studied, thus determining the general flow of wireless based attacks. This allows one to see which phases within the attack process are most critical to the attacks success and which are less important. The most critical phase was determined to be the probing phase. Thus it was given a weighting of 30% importance to the overall attack, whereas the least important phase and one usually ignored by wireless attackers the covering phase has been weighted at 5% importance to the overall attack. The actual determination of what percentage of the total assigned to each phase was based on the most important and least important phases. The values were then assigned accordingly by changing the values by increments of 5%.

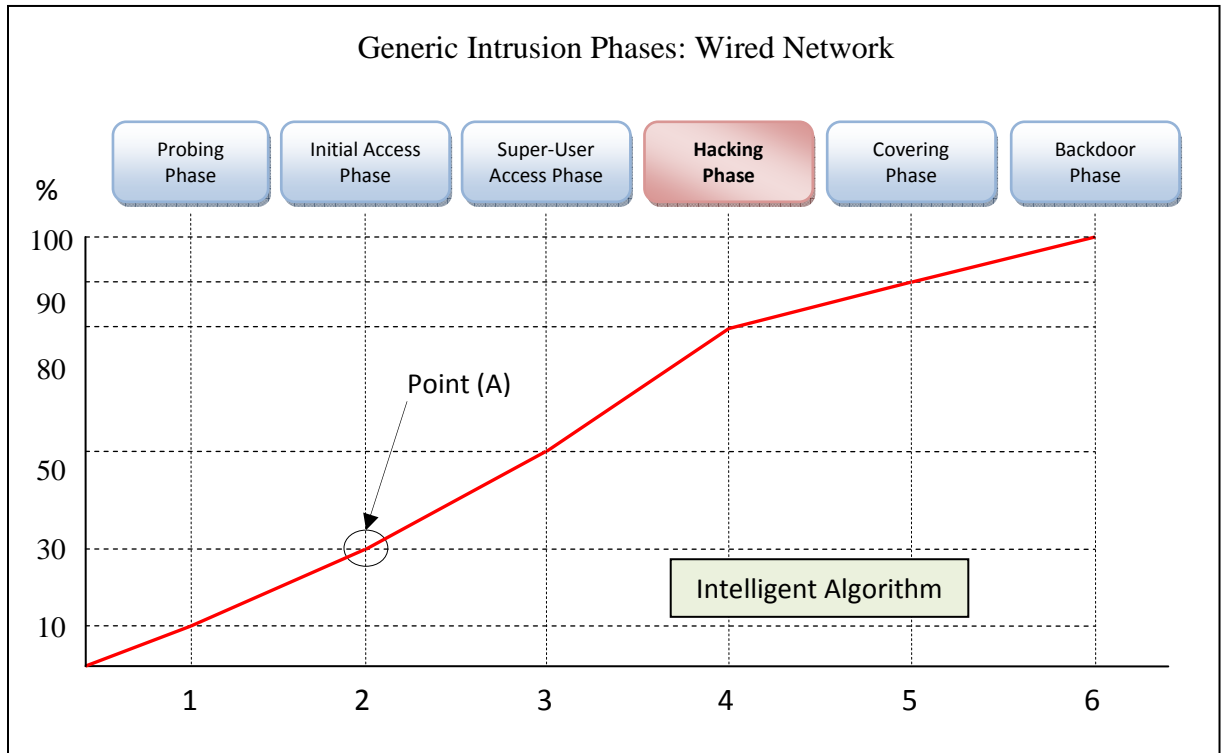


Figure 6.3: *Alternative Misuse Detection Approach (Wired)*

Figure 6.3 is a graphic explanation of the alternative misuse approach, focusing on the wired network portion of the approach. The figure shows the six generic intrusion phases, each consisting of multiple generic signatures and the intelligent algorithm that together make the alternative misuse approach.

The alternative misuse approach gathers precise data from various information sources. This information includes audit log information and various user profiles. The following simple example illustrates the operation of the alternative approach as it functions within a wired network environment: There is evidence of a user probing the network, via the wired network, with a port-scanner in the firewall logs (probing phase signature). Including the fact that there were illegal firewall access attempts (initial access phase signature), these attempts occurred after working hours (initial access phase signature). It can be predicted with relative certainty, by looking at the graph that the user is in the process of performing a theft of information intrusion attack.

The intruder/user has already gone through the probing and initial access phases, giving a certainty of 30% (10% + 20%) probability of attack to the particular user through his actions up till this point. This means the system administrator can be 30% certain that the user is in the process of performing intrusive activity over the wired network (Point A).

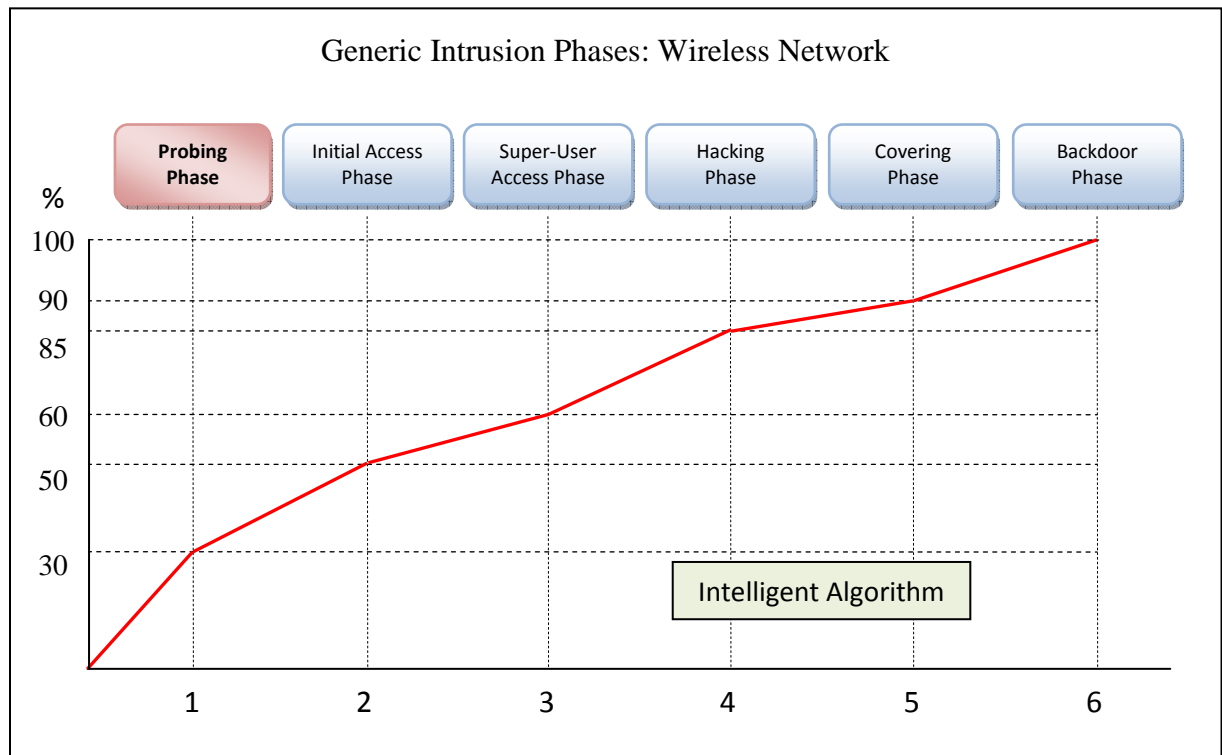


Figure 6.4: *Alternative Misuse Detection Approach (Wireless)*

As opposed to the previous example (showing the operation of the alternative misuse approach as applied to a wired network environment), the rest of this section shows how the alternative approach differs in weightings for attacks over an organization's wireless network.

As has been stated previously in Chapter 3, wireless networks are not as easy for an administrator to secure, due to their broadcast nature. With this in mind, one can see that the weightings put on the probing of wireless networks, as well as initial access, are of far greater importance than those same phases on a wired network, where the administrator

has greater control over the security. The information gathered comes from various sources around the wireless network, including information from wireless hosts and wireless access-points. This precise information allows for the accurate determination of user activity on the wireless network and allows for precise determination of activity within the critical probing and initial access phases.

A simple example of how the updated alternative misuse approach weighting scheme works on a wireless network is as follows: if evidence was gathered from a wireless sensor that a user/intruder has probed an access point or wireless host (probing phase signature), then the system starts to pay more attention to the user/intruder's next activities. If evidence has been detected previously that there also exists data that a rogue access-point (an access-point setup by a hacker to mimic an organization's access-point) has been used to gather login credentials when users log into it (probing phase signature), and if no other probing phase signatures are fired, the system continues to look for evidence in the initial-access phase. Along with the previously mentioned data, if there also exists evidence that a user has had multiple login failures (initial access phase signature), or has attempted to connect to the network with an incorrect WEP key (initial access phase signature), then the probability of the particular individual performing an intrusion attack against the wireless network is calculated as 50% (30% + 20%).

This means that the system administrator can be 50% certain that there is an intrusion attack taking place on his/her network. They can, thus, take the appropriate measures before the attack escalates. The reason the system and, thus, the administrator knows the actions are part of the same attack and are being performed by the same user, is the fact that when the user/intruder first logs onto the wireless network, the user/intruder's MAC address is logged. As he/she progresses through the attack, the system follows the individual, based on the MAC address.

If the intruder does change the MAC address, then the second engine couples the action to the same user/intruder through artificial intelligence. The next section gives an overview of the fuzzy methodology used to implement the updated alternative misuse approach.

6.2.2 Fuzzy methodology

As mentioned previously, this section focuses on the fuzzy methodology and fuzzy logic used to implement the updated alternative misuse detection approach and the NeGPAIM-W fuzzy engine. During initial research done at NMMU (Botha, 2003), many technologies were considered in the implementation, including the Dempster-Shafer Theory of Evidence, Bayesian Technique, Certainty Factors and Fuzzy Logic. Fuzzy logic was finally settled upon, through intense research, as the best method for implementation of the original alternative misuse detection approach. As the updated alternative misuse detection approach is based on the original alternative misuse detection approach, fuzzy logic still applies and is used.

The methodology works as follows. An intelligent algorithm, based on fuzzy logic theory, creates graphs (each graph represents the actions of the user/intruder) by constructing triangles. These graphs are then compared against each other, and the resultant output is discussed later in this section.

The objective of the methodology is two-fold:

1. Firstly, to interpret the input data received from the various sources (sensors); and
2. Secondly, to interpret the combined data according to a transfer function. This is done by creating and comparing two graphs.

Two graphs are created by utilizing the above fuzzy logic methodology. These graphs are then be compared by using a pattern-recognition technique. The first graph is called the **Template Graph** and is a representation of the authorized actions of the user / intruder on the system in terms of the six generic intrusion phases. The second graph represents the actual actions of the user or intruder in terms of six generic intrusion phases and is known as the **User Action Graph**.

Every time a new user / intruder is “discovered” on the system; a unique template is created and stored for that particular user. The template is based on the specific user’s rights and privileges, as recorded in the user profile for that user. In the case of an unknown or unidentified user, a standard template is used. The user action graph is dynamic and

represents the actual actions of a user on the system. A user action graph is constructed as soon as the template is stored in the database. The user action graph is updated every few seconds, and it is based on the input data collected from the various sources. The user action graph is also represented in terms of the six generic intrusion phases.

As soon as the user action graph is constructed, the fuzzy methodology starts searching for intrusion patterns. This process is conducted by mapping the two graphs onto each other. The mapping process is represented by the sample graphs in Figure 6.5. Figure 6.5 (a) represents the template to be constructed for a new user, based on the specific user's rights and privileges. The template is in the form of six combined triangles. Each triangle represents one phase of the generic intrusion phases. In practice, more than one triangle can be used to represent one single phase, but for the purpose of describing the methodology, only one triangle represents a single phase.

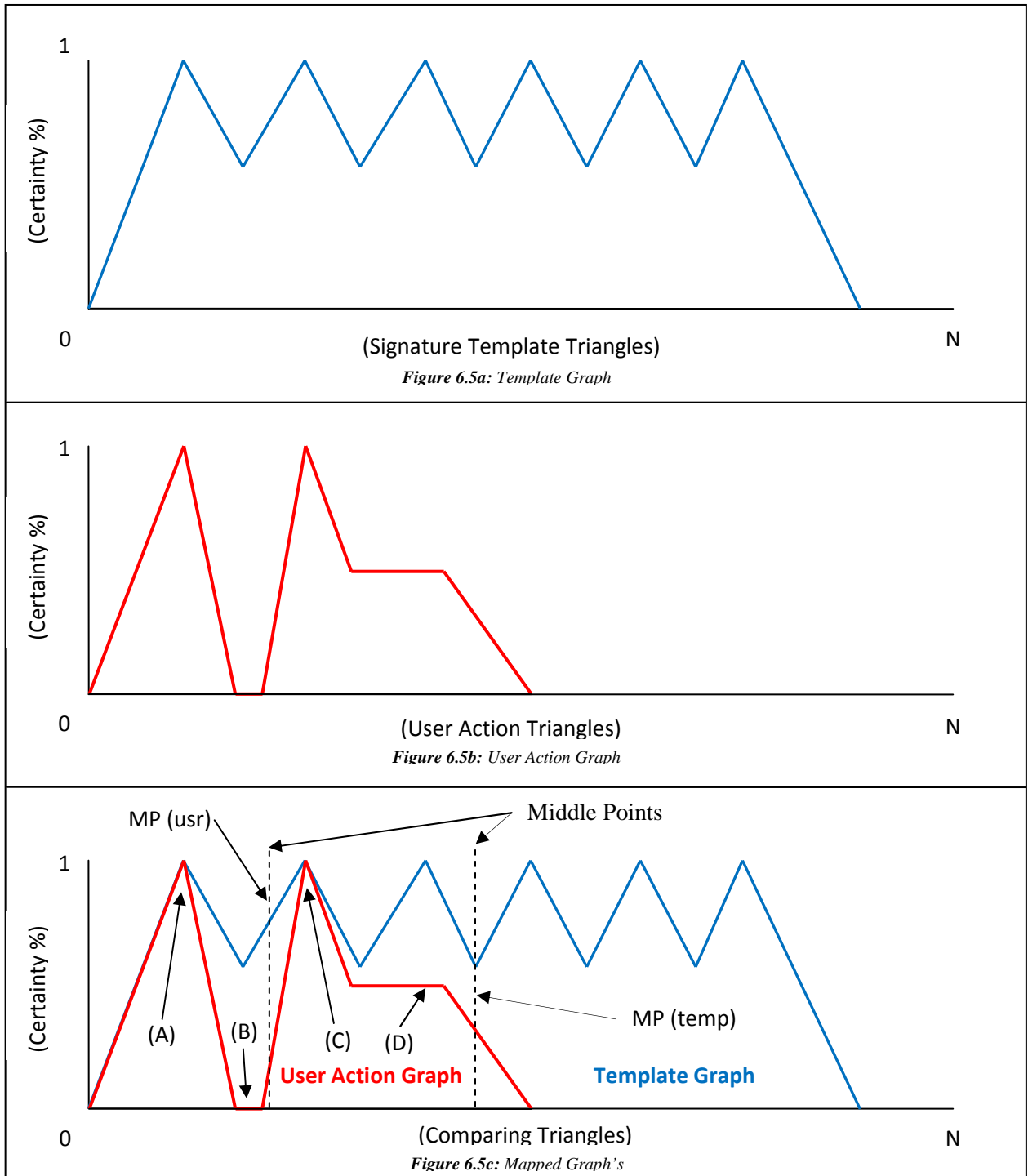


Figure 6.5: Sample Fuzzy Graphs

Figure 6.5b represents a typical user's action graph. The graph also consists of several triangles, each numbered from 1 to N as is with Figures 6.5a and 6.5c. Each numbered triangles represents the results of one of the generic intrusion phases. The shape of each triangle is determined by the certainty that a user/intruder has completed one of the generic intrusion phases. In the example in Figure 6.5c, enough evidence was found to be certain that the user completed phase one of the series of generic intrusion phases; therefore, the shape of the triangle (A) shows 100% certainty when mapped against the template. No evidence could be found to indicate that the user is busy or has completed phase two; therefore, no triangle (B) has been drawn. Enough evidence was found to be 100% certain that the user did complete phase three of the series of generic intrusion phases that resulted in the second full-size triangle (C). Some evidence was found to indicate that the user is busy or completed a part of phase four. This lack of evidence indicates that the certainty factor for this phase is about 50%; therefore, the shape of the triangle (D) is half size. No further evidence could be found to indicate that the user is busy or has completed phases five and six. This resulted in a straight line.

Figure 6.5c shows how the two graphs are compared or mapped on top of one another. The mapping of the two graphs is conducted by calculating the area of each graph. By comparing the two middle points of each graph, namely $MP(usr)$ and $MP(temp)$, one can determine whether or not the two graphs are similar (Berkan, 1997; Kosko, 1993). If they are not similar, the methodology determines how closely the two graphs are matched and, thus, how far the user / intruder has moved through the six generic intrusion phases.

There are a few areas of concern with the above mentioned method. First is the fact that the method is not proactive and this is one of the main reasons for this research. Secondly, this method does not provide an output that is meaningful to an administrator looking at the results. For these reasons, there is a need to consider an identification methodology that is proactive in nature, allowing the system to predict the next action an intruder will take, based on his previous actions. This proactive methodology is introduced next by means of the dynamic proactive identification model, which can be used by the fuzzy methodology to combat intrusions in a proactive manner.

6.2.3 Dynamic proactive identification model

The previous section introduced a method of mapping graphs on one another to check for intrusion attacks. It concluded that there is a need for a proactive identification model, and this model is now introduced. The model is based on the alternative approach, implemented through fuzzy logic. The objective of this method is to provide a detailed explanation of how the intrusion attack is performed and is based on the following two concepts:

1. To provide a detailed explanation of an intrusion attack, be it on a wired or wireless network. One has to follow an intruder while he/she is moving through the six generic phases, as previously mentioned, paying close attention to the probing and initial-access phases in the case of a wireless attack. Information gained by following the intruder can then be interpreted and used by the system administrator and/or the system to perform various active responses. Such responses could be disconnecting the intruder from the system and blocking his MAC address from accessing an access-point in a wireless attack.
2. To implement this method one, firstly, has to identify the generic intrusion phase that was reached by the intruder and, secondly, to predict the follow-up action(s) to be carried out by the intruder. These can be predicted, based on the type of network on which the attack is taking place. On wireless networks, attacks usually commence with scanning for a network and once found, a connection is attempted. The attacker then usually attempts to log onto the domain. It is, thus, possible to predict some or all the activities of an attacker, based on his prior actions.

To identify which phase was reached by the intruder is relatively easy when analyzing the fuzzy rules activated. For example, if only the illegal probing request and illegal monitor mode fuzzy rules have been activated, then one can make the assumption that the intruder has only, at most, completed phase one of his/her attack, and that he/she will more than likely be moving onto a next phase, involving activities, such as attempts to log onto the wireless network, and then attempts to log onto the organizational domain. If he/she does not have the WEP key, he/she will usually attempt to crack it. If the intruder fired fuzzy rules in more than one phase, one can determine which rules were recently activated and use this

information to determine the phase that is currently being conducted by the intruder, based on this information gathered from the fuzzy engine.

As intruders do not always think in the same manner, it is not as easy as one would think to determine the next step in an intruder's attack. A good example would be when an attacker, attempting to attack an organization over the wireless network, once he/she has logged into the network and gathered network information, he/she logged off and attacked the network, via the wired network instead, in an attempt to fool the system. For this reason, the dynamic proactive identification method needs to monitor the remaining phases by analyzing the input variables and, more importantly, the relationship between the inputs. All user information gathered about a potential intruder is also taken into account, including their MAC address and computer name. According to intensive research done at the NMMU (Botha, 2003), to obtain a clear picture of this relationship between the inputs (thus to obtain more detail on the intrusion attack), one can use a **fuzzy cognitive map**.

A fuzzy cognitive map (FCM) uses a symbolic representation for the description and modeling of a system. FCMs utilize concepts to illustrate different aspects in the behaviour of a system, and these concepts interact with one another, showing the dynamics of the system (Stylios et al., 1997). Although an FCM is constructed for each of the six generic intrusion phases, the rest of this section will focus on the FCM of the initial-access phase, thus, illustrating the concepts of the dynamic identification method, and in so doing, the FCM functionality. Figure 6.6 shows the FCM for the initial access phase of a wireless-based attack. This FCM consists of five nodes, also referred to as concepts, and six edges. Each of the five nodes represents a single intrusion event. The edges describe the relationships between the nodes (intrusion events). The edges also indicate whether one event increases or decreases the likelihood of another intrusion event (Stylios et al., 1997; Botha, 2003). It must be added that all the values in the graph are fuzzy and, thus, take arguments in the range of $[0,1]$ and the weightings on the arcs, which are in the range of $[-1, 1]$, indicating the degree to which each event affects another. This can be seen in Figure 6.6 (Stylios et al., 1997).

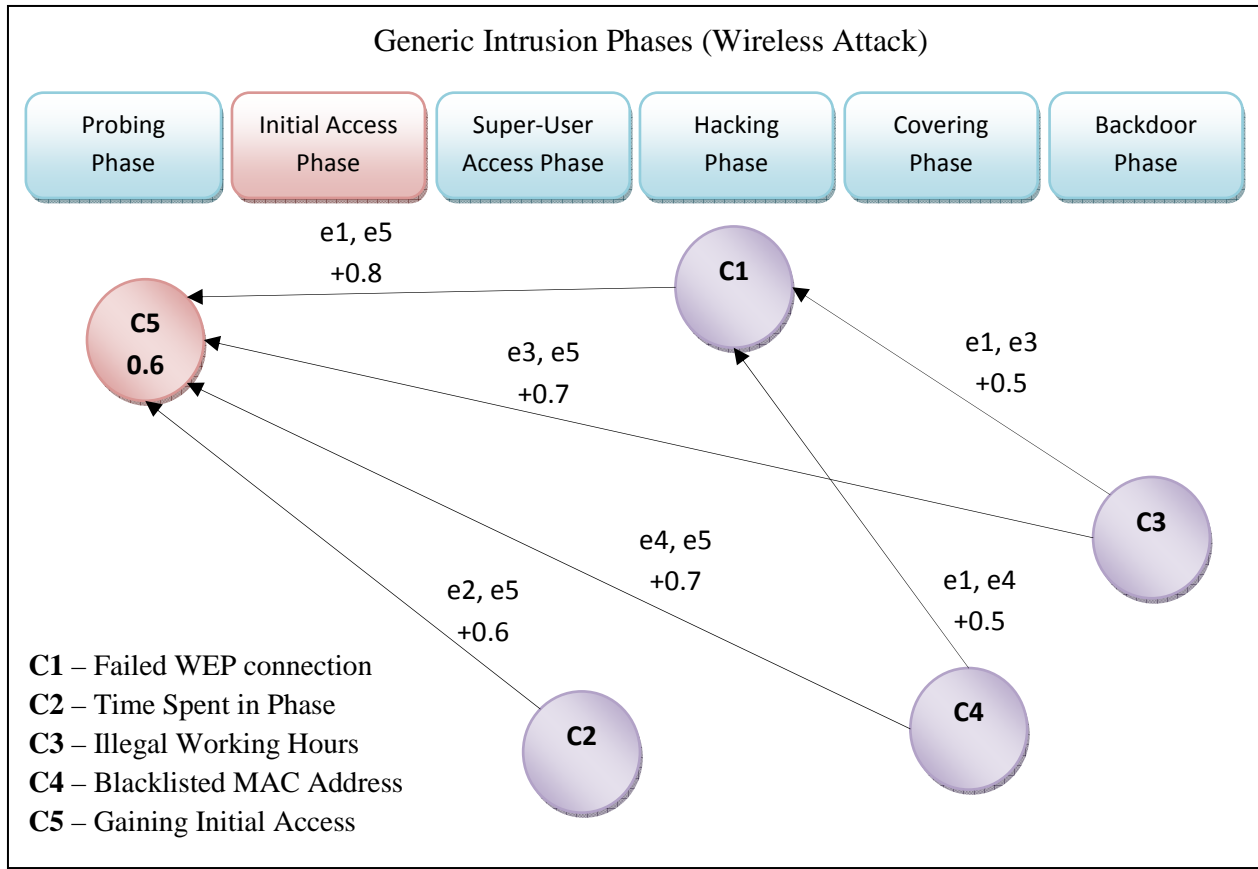


Figure 6.6: FCM for Initial Access Phase of a Wireless Attack

In the figure, one can see the relationship between the events, for example, the relationship or ark between C4 and C5 also known as (e4, e5). With this relationship, there is a positive relationship of value 0.7 between the two nodes, implying that if the number of failed WEP connections increases, the possibility that the attacker is still in this phase increases by a degree of 0.7 or 70%. These values have been calculated using the data contained within Table 6.2.

The dynamic identification method is based on the FCM concepts and the method is performed in four basic steps. The steps are as follows:

- Firstly, to determine whether an intrusion event did take place;
- Secondly, to determine the incoming relationship value for the different events;
- Thirdly to determine whether the phase was fully conducted by the intruder; and
- Lastly, if not fully conducted, identify the possibilities that the various events might take place in the future and inform the system administrator.

This section highlighted the inner workings of the fuzzy engine component of the NeGPAIM-W Model, one of the three engines to be discussed in this chapter. The next section explains the second of the low-level engines, namely the neural engine. The neural engine operates simultaneously with the fuzzy engine. Where the fuzzy engine utilizes fuzzy logic to perform its misuse detection, the neural engine makes use of artificial neural networks to perform anomaly detection. These include generating user footprints and determining whether or not any user has deviated from his profile.

6.3 The Neural Engine

As concluded previously, the neural engine operates simultaneously with the fuzzy engine. The purpose of the neural engine is to complement the fuzzy engine. This section introduces the neural engine and the neural methodology, as well as explaining how the neural engine can complement the fuzzy engine. One of the purposes of this section is to clearly indicate the differences between misuse detection and anomaly detection, as well as to show how they can complement one another. There is a clear distinction between misuse detection and anomaly detection in the sense that generally, misuse detection cannot identify new attacks as stated in, but anomaly detection can. As can be seen from the previous section and the introduction to this section, misuse and anomaly detections have their own advantages and disadvantages.

Misuse detection focuses on detecting attacks that it has listed in its detection database, and as mentioned before, it cannot detect new attacks. Anomaly detection, on the other hand, complements misuse detection in that it is geared more towards detecting new and unknown attacks by detecting abnormal behaviour. This allows for a greater range of attack detection: if the fuzzy engine does not detect an intrusion, it will usually be detected by the neural engine and vice versa. Thus, the neural engine complements the fuzzy engine by firstly searching for abnormal behaviour on the system and secondly, by linking all user actions on the system to the responsible user account within the organizations directory service, so the actions can be traced to a single individual.

An example of this could be an intruder, who gets user login credentials and passwords for users A and B. User A's credentials are used to steal the password file. In the case of user B, the intruder uses his credentials to gain access to critical information on the system. The IDS needs to have the ability to know that the intrusion, although performed using two different accounts, is actually the same user, in this case, the intruder. This section will also address this problem and explain the inner workings of the neural engine. This will include the neural engine's implementation of the anomaly-detection approach to intrusion detection.

The neural engine will be explained in terms of the following topics:

1. Neural networks methodology; and
2. User identification strategy.

These two points will form the basis for this section and will enable one to better understand the internal workings of the neural engine.

6.3.1 Anomaly Detection Through Neural Networks

As indicated above, neural networks have huge potential to detect intrusion attacks, and it makes a lot of sense to use them in the implementation of the NeGPAIM-W anomaly-detection engine. The previous NeGPAIM model made use of artificial neural networks, and as the crux of the neural engine still remains the same, artificial neural networks are still used in the implementation of the updated model's neural engine. With the background on anomaly detection from Section 2.5.2, and a method to implement the engine, namely neural networks, it is now possible to define a methodology that will implement the neural engine. This methodology's purpose is to detect abnormal user behaviour on the system. This section focuses primarily on the methodology and changes to it as it applies to the detection of wireless-based attacks.

The NeGPAIM neural methodology was based on the following assumption:

"Each user on the system is unique and leaves a unique footprint on a computer system when using it. If a user's footprint does not match his/her reference footprint, based on normal system activities, the system administrator or security officer can be alerted to a possible security breach." (Botha, 2003)

As the updated NeGPAIM Model (NeGPAIM-W) is concerned more holistically with network information, as well as system information in combination, the NeGPAIM-W neural methodology is based closely on the following assumption:

“Each user on the network is unique and leaves a unique footprint on the network and computer systems it supports. If a user’s footprint does not match his/her reference footprint, based on normal network and system activities, the system administrator or security officer can be alerted to a possible security breach, including the source network of the breach.”

For this assumption to be correctly implemented, a footprint of each user needs to be defined. For the purposes of this dissertation, the footprint of the user is defined as the total behaviour pattern of a user when interacting with a network and any connected computer systems. The total behaviour pattern of user interactions consists of three parts, that is, the behaviour of the user, the behaviour of the computer system and the behaviour of the network. Examples of metrics that can indicate the behaviour of the user are:

- (i) the set of typical commands being used by the particular user;
- (ii) the frequencies with which they are being utilized;
- (iii) the packet size;
- (iv) the bandwidth utilization of the user; and
- (v) the type of network utilized.

The behaviour of the system can be defined in terms of the system responses to the user behaviour. An example of metrics that can indicate system behaviour response is:

- (i) if the user is allowed to use a network application, such as trace-route or FTP, the memory usage, processor power and network utilization for the application can represent the behaviour of the computer system.

The behaviour of the network can be defined in terms of the network responses to the user behaviour. An example of metrics that can indicate network behaviour response is:

- (i) if the user has permissions to access the wireless network, the user’s bandwidth consumption, protocol types, packet sizes and number of connections open.

The total behaviour pattern of users must also include the needs of every user on the network and computer system. Take for example, some users may make use of the computer system and network to send and receive e-mail, whilst other users, such as sales people, may transfer large amounts of information across the wireless network, while logging orders. Therefore, the needs of every user should be directly proportional to the time spent by the specific user on the network and computer systems. For example, the time the wireless network link is utilized by a salesperson should not be longer than is needed for him to perform his tasks and is, therefore, role-based.

Figure 6.7 below represents the total behaviour pattern as it is seen in a diamond 3D diagram. This diagram shows the relationships between the components. The figure below represents the relationships between the user behaviour pattern, the system behaviour pattern, the network behaviour pattern and the user needs pattern.

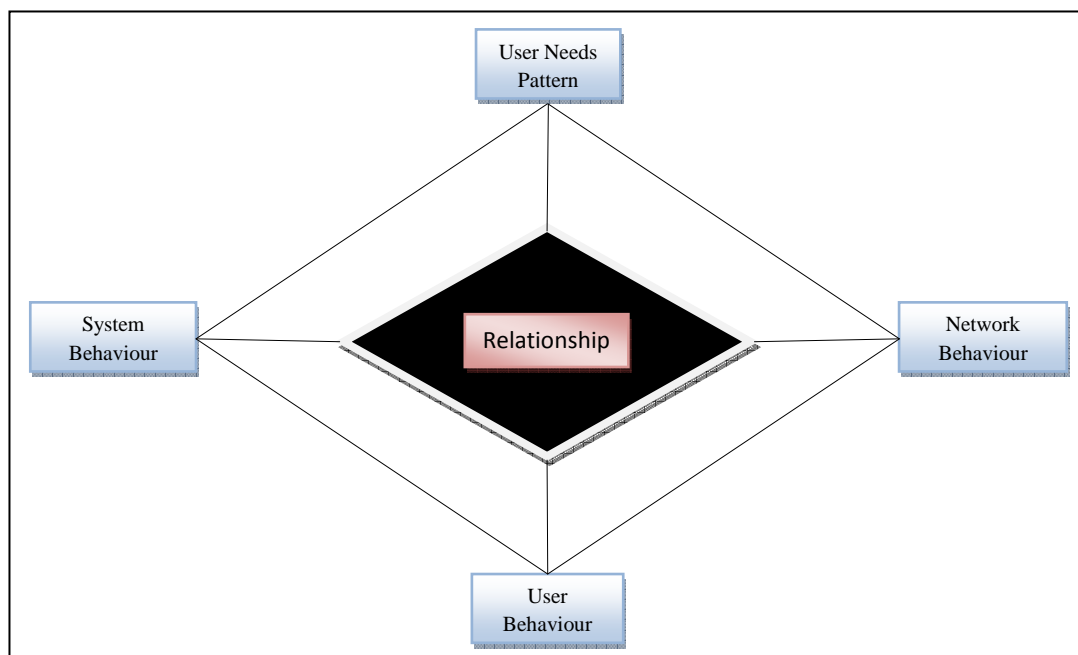


Figure 6.7: Relationship Diamond Model

An example of a relationship between the user behaviour pattern and the network behaviour pattern can be represented by the time interval between a user attempting to access a network, and the network itself responding to that access attempt. The same is also true between the user behaviour and user needs' patterns. This can be represented by the time of day the user

attempts to access the network resource and the time that the user needs the access to the network. This time interval should be within the user's normal working hours.

The above methodology explained how the neural engine determines whether a user is performing normal or abnormal activity on the system. As was discussed, the user's actions can be identified as normal or abnormal by a total behaviour pattern. This pattern consists of four relationships between the user behaviour pattern, the user needs' pattern, the system behaviour pattern and the network behaviour pattern. This allows the neural engine to analyse the users' actions as they apply to the various patterns. Next is an explanation of how the user is identified and, thus, linked to the actions he/she performs on the system, be it normal or abnormal activity.

6.3.2 User Identification Strategy

The main purpose of the neural methodology is to do anomaly detection. The second purpose is to identify the user performing multiple actions under different user names. In Section 6.2.3, it was indicated that the fuzzy methodology's effectiveness in detecting intrusions hinges on its ability to link every action on the system to a specific user account, and thus, the person responsible for the action. In accomplishing this goal, a strategy must be defined that implements the following two steps:

- Firstly, it must determine whether the action is performed by a registered user on the system, and, if not;
- It must secondly, construct a historical user behaviour profile for that new user and utilize this profile in conjunction with the rest of the profiles to couple future user actions on the system to a registered user name.

To provide more clarity on this strategy, consider the following scenario: *System A* implements the user identification strategy. It constructs reference patterns for each user on the wireless network, based on their MAC address, similar to the historical behaviour pattern explained in the previous section. The system then monitors all actions performed by the users over the wireless network and couples a user name and MAC address to each of the actions performed, where a user action could be sending e-mail or typing a document, etc. After a while, it detects a user action that cannot be traced to one of the registered users, and it

constructs a new reference pattern for that user. This user has the same MAC address as the user previously mentioned. The system calls the user, “*user Z*”, and links it to the MAC address. Thereafter, it detects another new user action and determines that this action also corresponds to the actions performed by *user Z*, but the MAC address of the node used to perform this action has changed. Without this user identification strategy, the system would not be able to couple the unknown actions to a specific user and computer. More detail on this strategy is provided in Chapter 7.

Section 6.2 dealt with the implementation of the fuzzy engine component, and this section has dealt with the neural engine component of the NeGPAIM-W Model. Now that the two low-level processing engines have been fully explained, the next section explains the high-level processing engine. This high-level engine, as stated previously, is the central analysis engine (CAE), which uses statistical calculations to combine the output from both the neural and fuzzy engines. This combined output allows for a more holistic description and understanding of the intrusion attack taking place.

6.4 The Central Analysis Engine (CAE)

Sections 6.2 and 6.3 introduced the two low-level detection engines, namely the fuzzy engine implementing the misuse-detection approach, and the neural engine, implementing the anomaly-detection approach to intrusion detection. The output of these two low-level engines needs to be correlated so that the two engines can assist one another and, ultimately, determine the overall intrusion status.

The central analysis engine (CAE) does just this. By implementing statistical calculations and concepts, the CAE combines the output of both the fuzzy engine and the neural engine. The resultant output of the calculations done on the fuzzy and neural outputs is known as the total intrusion probability. This total intrusion probability will indicate whether or not the user/intruder has, in fact, performed an intrusion attack or not. It also allows the CAE to determine whether any active or passive responses need to be implemented.

6.4.1 Functions of the CAE

As was previously stated, the central analysis engine is the high-level processing component of the NeGPAIM-W Model and, as such, does not do any direct intrusion detection by the utilization of either anomaly or misuse detection. Rather, this component is responsible for analyzing the outputs gained from the two low-level components, namely the fuzzy and neural engine. The CAE's functions can be grouped into two categories: primary functions and secondary functions. The primary and secondary functions have been listed below in the appropriate categories.

Primary Functions:

- Combines misuse and anomaly intrusion values from low-level engines. This is done to gain perspective on the overall attack and to allow the correct responses to be fired.
- Interprets the combined intrusion values, converting the mean value into a percentage probability of attack between 0 and 100%.

Secondary Functions:

- Interacts with the internal manager a component which allows the IDS to store intrusion and configuration information e.g. logging attacks to database, determining the implementation of both configuration and security management.
- Interacts with the external manager a component which allows the administrator access to the IDS through its graphical user interface (GUI), configuration of active and passive intrusion responses.
- Interacts with the internal responder a component residing on client machines allowing actions to be taken against an intruder, by the use of active responses.
- Interacts with the external responder a component residing on the IDS server which reports intrusion information as it occurs, management and interaction with the external responder, which runs various passive responses
- Provides storage capability, allows management and interaction with the central database to store data on intrusion events.

Now that the most important functions of the CAE have been explained, one can now begin to explain the essential statistical calculations that will be used to implement the CAE. The

purpose of the next section is only to give a short background on the statistical calculations, thereafter indicating how they will be implemented by the CAE to perform the primary and secondary functions.

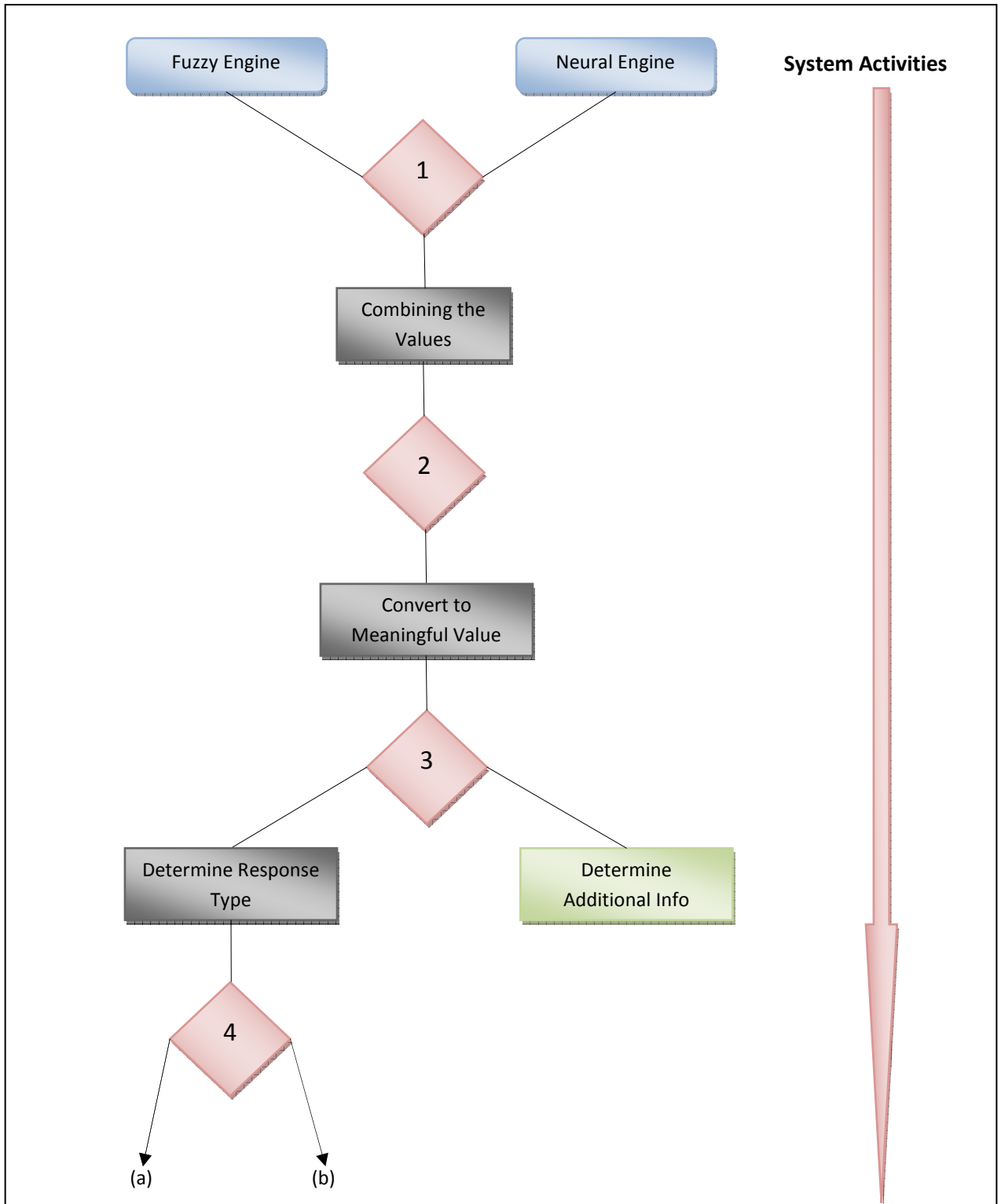
6.4.2 The central analysis methodology

As mentioned previously, the main function of the CAE is to provide an overall intrusion attack probability rating. This is done by performing statistical calculations on the output of the two low-level detection engines, known as the fuzzy engine and neural engine. The method of calculation done on the output of the two low-level engines is known as the descriptive statistic technique. It is a commonly used technique to combine values of two or more outputs, in this case the fuzzy and neural engines, and it is also used to do interpretation of the resultant output, making it more accurate.

The calculation used for the NeGPAIM CAE is the same for the updated model NeGPAIM-W, as the core idea of NeGPAIM-W has not changed. The basic calculations used to combine the low-level engines outputs have been depicted below in Figure 6.8. The calculations shown in Figure 6.8 form the basis of the methodology behind the CAE, known as the central analysis methodology. The central analysis methodology is based on the flow chart seen in Figure 6.9.

$\text{Arithmetic mean} = \frac{\text{Neural Output} + \text{Fuzzy Output}}{2} \quad \dots(1)$
$\mu = \frac{\sum x}{n}$ <div style="display: flex; justify-content: space-between; align-items: center;"> <div style="flex: 1;"> <p style="margin: 0;">To convert this statistical means into an intrusion probability value, the expression below is used.</p> </div> <div style="text-align: right;"> <p>... (2)</p> </div> </div>
$\text{Probability Value} = \mu \times 100\% \quad \dots(3)$

Figure 6.8: Example of Calculation for Gaining Total Intrusion Probability (Botha, 2003)



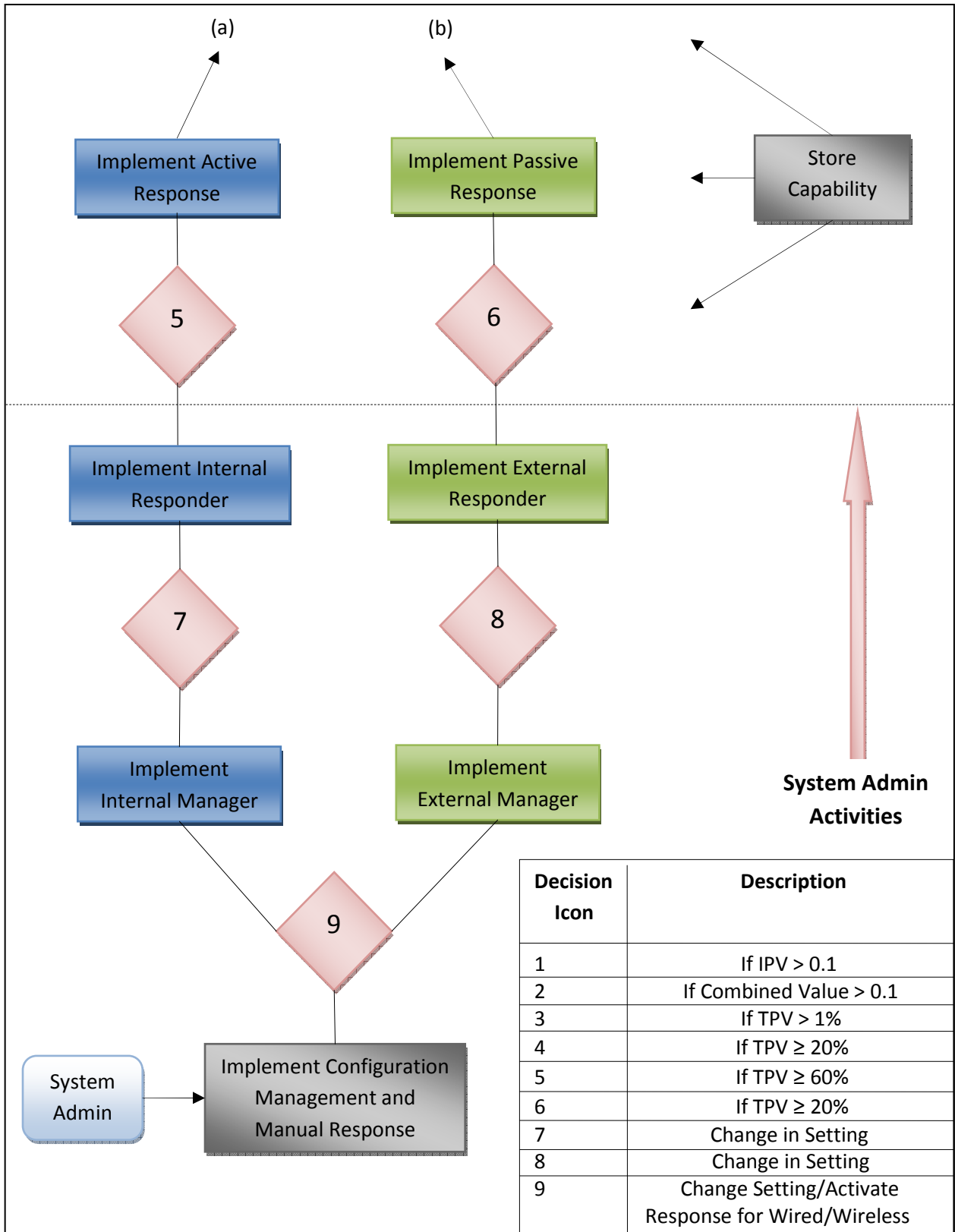


Figure 6.9: Flow Chart of Central Analysis Methodology (Botha, 2003)

The methodology is divided into two separate and distinct sets of activities listed and discussed below in Sections 6.4.2a and 6.4.2b:

6.4.2a System Activities

The first set of activities refers to the system activities and represents the actions that will be performed by the engine itself. As was mentioned previously, the CAE does not detect intrusion attacks directly, but rather it activates when it receives intrusion probability values from the low-level engines. As soon as it has this information from either/both of the low-level engines, it commences with the calculations in Figure 6.7. The engine will only calculate a probability value when a set of ten values has been passed to it. This is because the sample size for the calculation is ten.

The first function of the CAE is to combine the intrusion probability values (IPVs), as seen in activity 1, on Figure 6.9. The resultant output from this process is a combined IPV. The engine works on a First-In First-Out (FIFO) approach when calculating the statistical mean. This means when more than ten sets of input have been collected, it will only use the last ten and the other sets will be discarded. The reason for this is because the engine can receive a lot of information to process in a short time, and if the scheme was not implemented, the engine would have a bottleneck while processing.

The second function that is performed by the CAE is that of interpreting the combined intrusion probability. This can be seen in activity 2 on Figure 6.9. The process will only occur if the combined value is greater than 0.1, the resultant output of this process is a Total Probability Value (TPV). TPV is reached by converting the combined intrusion probability value into a total or overall probability value that is presented in percentage form between 0% and 100%. Depending on whether the TPV is closer to 0% or 100%, the possibility that the user is performing an intrusion attack **increases** from little possibility to almost certain.

The CAE will also perform the following calculations if the TPV is greater than 1%.: the reason being that the administrator will understand the data better.

- The average of the statistical mean for the period will be calculated;
- The Intrusion Tempo will be calculated, indicating how many activities the intruder has performed in a set-time period; and
- The Intrusion Period will be calculated, from the time the first TPV was received, right up to the time of the last.

When the CAE calculates a TPV that is greater than 1%, it starts to determine whether a response is needed as a countermeasure to the intrusion attack. The type and exact response implemented are determined by which response would adequately address the intrusion.

Responses in the updated NeGPAIM-W Model also take into account whether the attack was detected on wired or wireless network when deciding which response to implement. The reason for this is the fact that different responses are implemented differently on different network media. Rather than cutting off all wireless access-points connected to the network when an attack is sourced as taking place on one of them, a better response would be to blacklist the attacker's MAC address on all access-points, via SNMP. Even if one needs to stop an attack sourced on the wireless network, the wired network will still be in operation due to the updates to the CAE. Previously, all network media would have had the same response implemented on them implicitly. This means that a severe wireless attack would have caused the wireless and wired LANs to have a lockdown.

With the updates now in place, only the afflicted network will be affected by the responses. The same is true for both active and passive responses. Passive responses to intrusion events will alert the administrator responsible for the network affected by the intrusion. In many large organizations, administrators are contracted to maintain specific parts of the network and, as such, when responses are fired, the primary administrator and administrator responsible for the network

segment will be alerted. This is so that intrusions and intrusive activity are seen as soon as possible. These changes to the operation of the CAE will allow it to cater better to larger networks.

Below one will find a list of some active and passive responses NeGPAIM-W is capable of performing:

Passive Response:

- Send alarm to console;
- Send e-mail;
- Send alert to pager, cell phone or PDA; or
- Send SNMP trap.

Active Response:

- Disconnect network connection;
- Disable access-point;
- Jam wireless signal;
- Reconfigure access-point/router via SNMP;
- Block nodes MAC address;
- Suspend user account;
- Disable user account; or
- Execute user-defined application.

All the above responses are stored in the central database and can be viewed and edited by the external manager component.

6.4.2b Administrator's Activities

The second set of activities refers to the system administrator's activities and represents the actions of the administrator to configure the model, in particular, the CAE. The CAE will allow the administrator to change various configuration and security settings as described in Section 6.4.1.

The central analysis engine will provide graphic user interfaces that will be used by the system administrator to make the changes and will then implement these

changes in real-time, including assignment of responses to the various network media. The central analysis engine will also allow the system administrator to activate active responses manually.

The engine will again provide graphic user interfaces (GUI) that will allow the system administrator to manually activate responses, and the engine will implement these responses in real-time. The GUI will alert the administrator about possible intrusion events. This will be shown in both a popup window that the administrator may click on to get more details and in the main GUI itself. Possible intrusive events will be displayed as neural probability, fuzzy probability and total probability of attack. Finally, the central analysis engine will allow the system administrator to view the recorded intrusion events and current configuration settings through the external manager GUI. This will allow him to view the health of both the wireless and wired networks to see whether anyone has attempted to infiltrate either. The administrator, through the GUI, will also have the ability to draw reports of all past and present network activity. This will allow him/her to further tweak the responses and network sensors to operate more efficiently.

To summarize the central analysis methodology, the methodology implements two sets of activities, that is, system activities and system administrator activities. This section explained how the statistical concepts are used to perform system activities, such as combining the intrusion probability values received from the two low-level components and interpreting the combined value.

The combined value is called the Total Probability Value (TPV) and represents the overall probability that a user or intruder is performing an intrusion attack. The engine then uses this value, in conjunction with the Intrusion Probability Values (IPVs) received from the two low level components, to determine which network and which responses to implement on the network.

The updated CAE also allows the administrator to assign both active and passive responses to different network media: in most cases, these media would be wired

and wireless network media. This allows network segments to continue data transmission even if one segment needs to be taken offline. In order to gain a better understanding of the CAE functions, the next section provides a practical example of an attack, focusing on the three engines and the detection process as a whole, including the engines' reactions to intrusion events.

6.5 Detection Example

The purpose of this section is to attempt to better explain how the three main engines of the NeGPAIM-W Model function by showing a practical example of an intrusion attack. This example shows how the attack is initially detected, all the way through to the passive responses alerting the administrator, and the implementation of various active responses. The attack chosen for this example scenario will highlight the newly implemented wireless detection capabilities. This attack will be explained in terms of the six generic intrusion phases. For the purposes of this example, the attacker will be known as *User X*, the intruder will be performing an attack over the wireless network of *Organization Y*, with the main objective to corrupt corporate information. *User X* will be using a notebook computer, with Linux installed as the operating system. The wireless network attack occurs as follows:

First of all, *User X* finds a place near *Organization Y* where he can stay out of sight of onlookers. *User X* then proceeds to change the mode in which his wireless network card operates to monitor mode, allowing *User X* to scan for wireless networks.

As *User X* starts to scan for access-points (Probing Phase), one of the generic wireless fuzzy engine rules fires. This occurs as *User X*'s wireless card's MAC address is not listed as known, and he/she is probing for the SSID of the access-point. The fuzzy engine notes *User X*'s MAC address, which it gains from one of the many probe request packets being sent to the access-point. At this point, the neural engines output is zero and the CAE's output is low. The output of the three engines would then be currently as follows:

- Fuzzy Engine = 15%
- Neural Engine = 0%
- Central Analysis Engine = 7.5%

User X receives a response from the access-point containing its SSID. *User X* now attempts to connect to the access-point using its SSID (Initial-Access Phase), but finds that it contains WEP encryption.

Another of the fuzzy engine's signatures fires at this point as *User X* failed to log onto the access-point successfully. This is correlated with the previous evidence collected, thus, starting investigation on *User X*. The neural engine at this point still has not found any evidence of anomalous behaviour, and the CAEs output is still low. The output of the three engines is currently as follows:

- Fuzzy Engine = 20%
- Neural Engine = 0%
- Central Analysis Engine = 10%

User X will now sit patiently, collecting enough data from *Organization Y's* wireless network to crack the access-points WEP key (Probing Phase).

Another fuzzy signature is fired. This is due to *User X's* long duration of sniffing traffic in monitor mode. *User X's* duration in monitor mode has been timed by the fuzzy engine since the first probe request. The misuse data is now fed to the CAE as a possible attack, the neural engine's output, thus far, is null, so the total probability of attack value generated by the CAE is still relatively low. There are no responses fired at this point, as there is not a greater threat. The output of the three engines is currently as follows:

- Fuzzy Engine = 40%
- Neural Engine = 0%
- Central Analysis Engine = 20%

At this point in the attack, *User X* has just gained the WEP key after utilizing collected data to crack the key. *User X* now uses this key with the SSID gained earlier to connect

successfully to the network. *User X* now attempts to brute force the password (Initial-Access Phase) of one of the employee's account's *User A* with a brute force tool.

Another fuzzy rule fires due to the multiple invalid login attempts by *User A*. The neural engine also detects a deviation in the normal behaviour pattern of *User A* as it is past *User A*'s normal work hours. *User A* also usually enters his/her password in correctly on first attempt and to brute force the password, the brute force application fails hundreds of login attempts before finding the correct password. The above information causes the fuzzy and neural engines to pass their intrusion provability values to the CAE for further analysis. The CAE determines that the total probability of attack is at 40%, causing a passive response to be fired. The output of the three engines and the system responses are currently as follows:

- Fuzzy Engine = 50%
- Neural Engine = 30%
- Central Analysis Engine = 40%
- Passive Response = E-mail administrator with warning.

User X uses *User A*'s username and his newly gained password to log into *Organization Y*'s corporate network. *User X*'s ultimate goal is to destroy data valuable to *Organization Y*. *User X*, thus, starts opening various folders on the intranet and scanning the network for other servers to which he/she can cause damage to (Hacking Phase). The neural engine now determines from these actions and the previous actions of *User A*, that *User A* is not acting as he/she normally does and passes its relatively high intrusion probability value once again to the CAE, which with the previous output of the fuzzy engine, determines that the total probability of attack is at 72.5% and fires an active and passive response. The active response may be to deny the MAC address used by the attacker on the access-points, and at the same time, to lock the user account *User A*, thus thwarting *User X*'s attempt to damage *Organization Y*'s data. The passive response would be to e-mail the administrator explaining to him/her what occurred and what countermeasures were implemented. The output of the three engines and the system responses are currently as follows:

- Fuzzy Engine = 75%
- Neural Engine = 70%
- Central Analysis Engine = 72.5%
- Passive Response = Page administrator with warning.
- Active Response = Disable MAC address.

6.6 Conclusion

This chapter focused on the three main engines of the NeGPAIM-W Model, two of which are low-level processing engines known as the fuzzy and neural engines. The fuzzy engine implements misuse detection as its detection method, and the neural engine implements the anomaly or behaviour-based detection technique. The fuzzy engine implements its misuse detection utilizing fuzzy logic, whereas the anomaly detection of the neural engine is implemented using neural networks.

The third main engine discussed was the single, high-level processing engine known as the central analysis engine (CAE). This engine utilizes statistical calculations to perform its functions, the main function of which is the combination of outputs gathered from the two low-level detection engines. The CAE also converts the mean of the low-level engine's outputs into a total probability value (TPV), which is in a form that administrators will better be able to understand, as it is represented in a percentage form.

The active and passive responses have also been introduced and discussed, including the updates to the way the CAE implements the responses. This was done to enable the separated response functionality for wireless and wired networks as they operate differently and, thus, require different responses. The main aim of the updates to the lower-level detection engines is to allow for faster and more accurate detection of intrusion attacks. This is evident on both wired and wireless networks, through the separation of misuse signatures and with the neural engine taking the network into account when building a user profile.

The next chapter focuses on practically implementing the updated model through various experiments. The experiments will be performed on wireless networks, including intrusion attacks identified in Chapter 4. These experiments show that there is actually a real-world problem with intrusions on wireless networks, and that NeGPAIM-W can successfully solve the problem and ultimately prove the validity of this dissertation.

Chapter 7

A Wireless Intrusion Attack Experiment

7.1 Introduction

As the use of wireless network technologies becomes more and more prevalent, the insecurities associated with wireless technologies have started to become a real problem. Earlier chapters, namely Chapters 1, 2 and 3, outlined various topics, such as the need for information security, risk management, real-world computer security problems and security mechanisms that might be used to protect an organization's information. The main focus of these chapters was to gain insight into current problems with security of information and to introduce intrusion detection and intrusion detection systems.

Chapter 4 introduced and discussed various attacks directly aimed at wireless networks. Most of these attacks are used to steal user information, in the case of an intruder setting up a rogue access-point, or gaining access to the rest of the network or computer systems, in the case of an intruder cracking Wireless Equivalent Privacy keys (WEP). Chapters 5 and 6 explained in detail the NeGPAIM-W Model, a theoretical solution to the ever increasing problem of protecting both wired and wireless networks from the barrage of intrusion attacks constantly being thrown at them.

This chapter attempts to prove, in a practical manner, that intrusion attacks are indeed a reality and can be easily carried out. The NeGPAIM-W's fuzzy engine is the primary focus of this chapter, and in particular, the updates to the original NeGPAIM fuzzy engine, allowing it to address the problems associated with wireless networks. The prototype also includes the neural and central analysis engines as secondary components. The experiment is discussed in terms of the following points:

- NeGPAIM-W Prototype;
- Prelude to the experiment;
- The experiment;
- The results; and
- Evaluation of the results.

The next section introduces the NeGPAIM-W prototype, implementing the three engines as described in Chapter 6. As mentioned previously, the focus of each of the following sections will be on the implementation and testing of the fuzzy-engine component.

7.2 The NeGPAIM-W Prototype

As mentioned at the end of the previous section, this section explains the NeGPAIM-W prototype that was tested during the experiments conducted, which is explained later in the chapter. Previously a prototype was created for the old NeGPAIM Model with extremely promising results. The only downfall of this model and thus, the prototype, is the lack of capability to detect attacks within a wireless environment.

With the tendency towards wireless network technologies being implemented in organizational networks for mobility and a host of other reasons, it is imperative that wireless-attack detection functionality be incorporated into NeGPAIM. The previous NeGPAIM prototype was a simple implementation of the previous NeGPAIM Model. The next section explains the new NeGPAIM-W prototype and the wireless detection capabilities added to the prototype.

7.2.1 Sentinel IDS

The objectives of implementing the NEGPAIM-W Model is as a proof of concept on the updates to the previous model and to show that an IDS, implemented in this manner, would produce far better results than a regular non-proactive IDS within both wired and wireless environments.

The need for an IDS that functions proactively without much user intervention is slowly becoming an indispensable part of any organization's arsenal of anti-attack software and hardware; the reason for this being the fact that more and more data is being stored in log files, the volume and veracity of attacks is out of control and it is ever increasingly difficult to keep the "bad elements" out.

As the capacity of both wired and wireless networks grows, the need for faster detection of attacks will become even more critical, as it becomes impossible to have the latest patches and service packs for all one's software.

Key elements of the NEGPAIM-W Model have been implemented in a fully functional prototype, named Sentinel IDS. These elements are namely the reporting fuzzy, neural and central analysis engines. The reason these elements were chosen is that they form the core backbone of detection and the feedback processes, allowing for the proactive detection of attacks. In order for one to fully understand how Sentinel IDS works, one needs a background on how the IDS is set up. The next section shows how Sentinel IDS has been implemented. This includes an explanation of the environment on which the IDS resides.

7.2.2 Implementation of Sentinel IDS

As mentioned in the conclusion of the previous section, this section firstly details the operating environment of the Sentinel IDS. Secondly it gives an explanation of an example configuration of the IDS. This allows one to gain insight into how the IDS operates and helps in explanations later in the chapter.

Sentinel IDS has been designed to run on Microsoft Windows 2000 / 2003 servers and Windows 2000 / XP Professional, utilizing a SQL Server database to store attack data and attack definitions. Responses, both passive and active, have been implemented as well as remote sensors (smart agents). The three engines namely the fuzzy, neural and central analysis engines (CAE) have all been implemented and are functional, as previously mentioned.

Sentinel IDS has been built using the Microsoft .Net framework version 1.1 so that it can function on multiple Microsoft platforms. The reason for the implementation being on the Microsoft Windows platform is because within Unix and Linux environments, access to information to perform intrusion detection is easily accessed, and the kernels of the operating systems can be extended at will, thus, making it easier to gain needed system information.

The opposite is true within the Windows environment, where it is not as easy to gain access to system information, the kernel is locked down and cannot be reprogrammed, and access is only via SDKs and APIs. For these reasons, the Microsoft Windows environment was chosen as the test bed for the NEGPAIM-W Model, although the model could be adapted to fit the Linux/Unix environments with little trouble. The layout and components of Sentinel IDS are shown below in Figure 7.1.

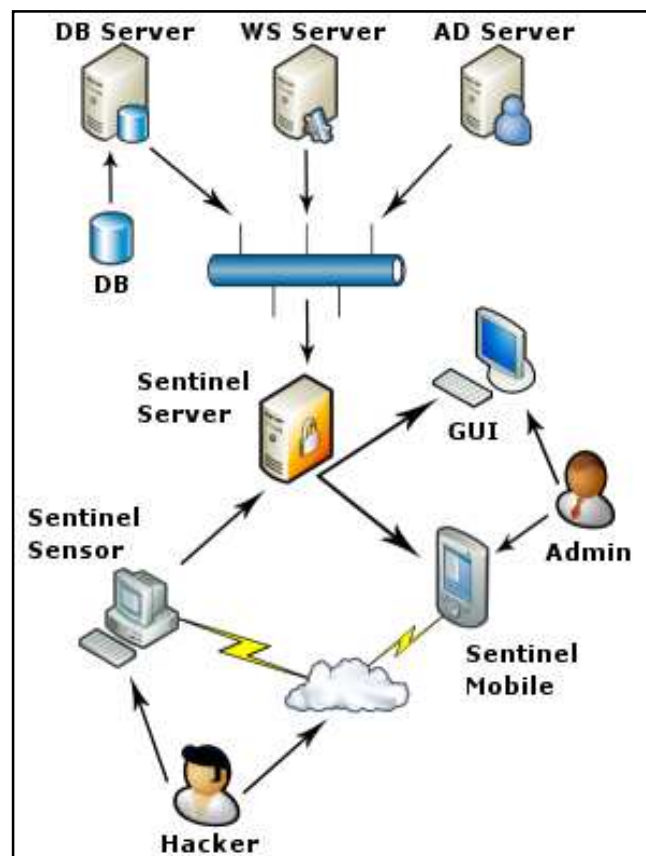


Figure 7.1: Sentinel IDS Layout

The Sentinel IDS server, as seen in Figure 7.1., has a direct connection to the organizational active directory, web-service and SQL-server database servers. This allows for optimal speed of access to needed information, and the system administrators have access to needed information, via the Sentinel mobile interface, for pocket PCs, as well as the regular Sentinel GUI, which contains a greater set of functionality.

The Sentinel mobile's functionality is a subset of the full Sentinel GUI's available reporting tools and works via web-service. The Sentinel sensors implement the NeGPAIM-W's fuzzy engine to save time on the collection of attack data. The reason for this is as a user connects, he is monitored by a sensor and the fuzzy engine detects abnormalities in his doings. As abnormalities are found, the fuzzy engine sends the data back to the Sentinel server to have its data correlated with data collected on other sensors from around the wireless and wired networks.

The neural and CAE engines are implemented on the Sentinel server. This is because these two engines need network-wide information both to determine the user's footprint, as is the case with the neural engine and to correlate and perform statistical calculations on the output of the neural and fuzzy engines, as with the CAE as seen in Figure 7.2.

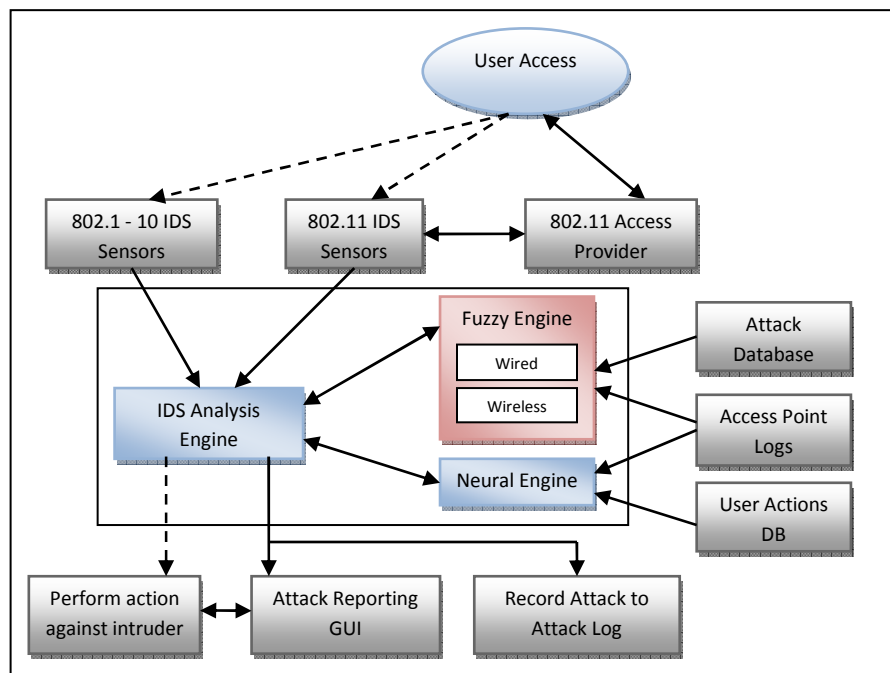


Figure 7.2: Low-Level Detection Model (Botha, 2003)

As previously mentioned, this chapter focuses mainly on the implementation of the updated fuzzy engine. The reason for this is that the fuzzy engine has undergone the most radical changes with regard to the detection and differentiation of wireless attacks, as opposed to its wired counterpart. The next section introduces and explains the updated fuzzy engine as it is implemented in the NeGPAIM-W's prototype Sentinel IDS.

7.2.3 Implementation of the Fuzzy Engine

The fuzzy engine, as described in Chapter 6, has been implemented in the Sentinel IDS in two parts, the first of which is the wireless and wired sensors that are placed around the network. The second part in the fuzzy engine implementation is the central fuzzy engine and signature database. This section, as with the rest of the chapter, focuses mainly on the wireless detection capabilities of the fuzzy engine.

The wireless fuzzy sensors are placed close by organizational access-points and constantly monitor packets destined for and dispatched from the access-points to which it is assigned. This is possibly due to the fact that the fuzzy wireless sensors have their interfaces placed in promiscuous mode. The reason for the sensors only monitoring organizational access-points is due to the fact that if one was to scan all wireless traffic, firstly, it would be invading the privacy of neighbouring organization's running access-points and secondly, the speed of detection would also be greatly affected if all data were to be continually scanned.

When the sensor is setup, explained in Section 7.2.4, the configuration contains a section that allows the administrator to set which SSIDs and MAC addresses to monitor, thus, only monitoring the data of interest to the organization and avoiding the problems listed above. Figure 7.3 is a diagram depicting the layout of the fuzzy engine.

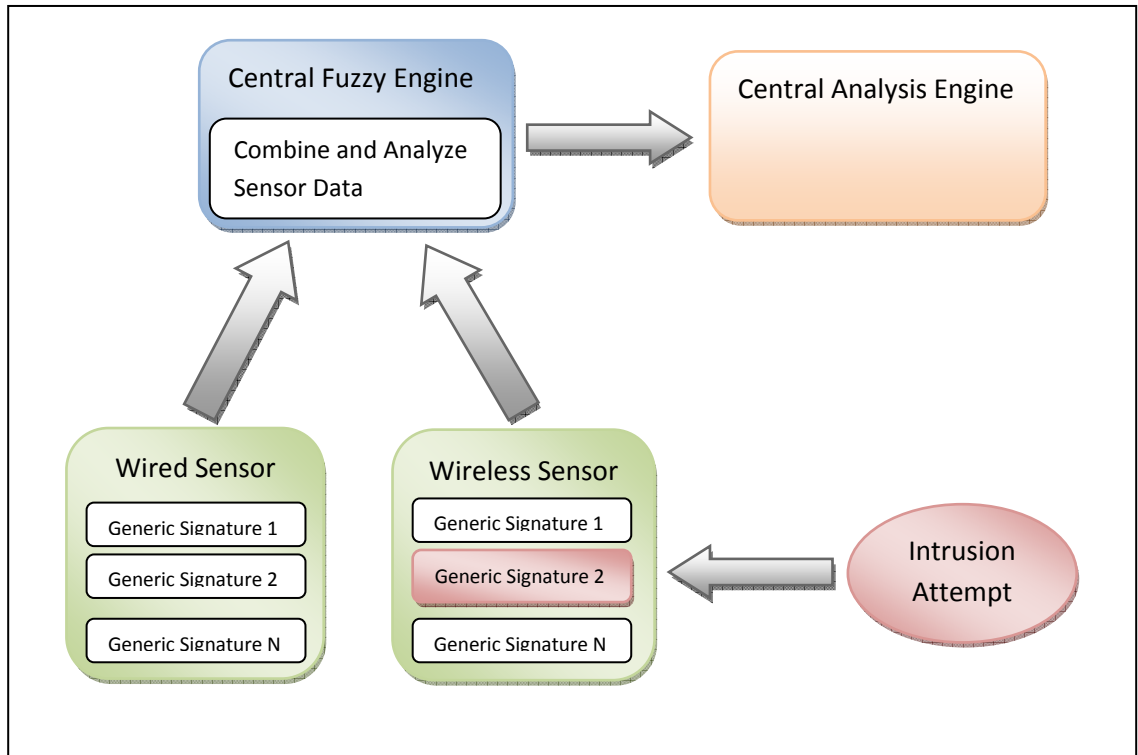


Figure 7.3: *Fuzzy Engine Components Diagram*

The wireless sensors, like their wired counterparts, have their generic signatures updated daily, hourly, etc., directly from the Sentinel server, allowing any new signature to propagate quickly and keeping the signatures on all sensors the same. As was mentioned in Chapter 6, the NEGPAIM-W Model segments the signatures into that of wired and wireless network signatures separately, conserving valuable processor time on the sensors which may be implemented on specially designed devices. This also allows the sensor to detect an attack occurring on the network segment in the fastest possible manner. As an attack is detected, the wireless or wired sensor reports back immediately to the central fuzzy engine. This is further discussed in Section 7.4.1.

Here, the intrusion or data that caused the sensor's signature to fire is logged to the database. Thereafter, the data is correlated with the data collected, via other sensors, both wired and wireless. This correlation of data by the central fuzzy engine gives a more holistic determination of whether there is an overall attack, which network segments are affected and the final fuzzy engine's probability of attack to be forwarded to the central analysis engine (CAE).

With the correlation of data on the central fuzzy engine, attacks aimed at both wired and wireless segments of the network can be examined and linked to the same individual. This was one of the reasons for the updating of the original NeGPAIM Model. This correlation also allows the CAE to implement responses, either separately, in the case that the attack is aimed at only the wireless segment or responses that can be simultaneous on both wired and wireless networks.

The next section explains the configuration of the fuzzy engine, as was referred to earlier in this section. This explanation is aided by the use of screenshots taken from the actual Sentinel IDS.

7.2.4 Fuzzy Engine Configuration

As mentioned in the previous section, this section discusses the configuration of the fuzzy engine, technical information on how the configuration works, as well as how the administrator would use the configuration. The configuration of the fuzzy engine is explained in a series of screenshots, so that it makes more sense.

Firstly, the configuration has been implemented in such a way that the administrator can save multiple versions of the configuration. This has been done by implementing the configuration files in xml format. This also allows the administrator to backup the configurations or edit them in an xml editor. Each of the fuzzy engine's two network sensors will have a configuration generated for them by the administrator on the server. These are implemented in the form of a questionnaire. This questionnaire will allow the administrator to tailor the specific sensor, be it wired or wireless, to the needs of his/her organization. An example of one of the sensor setup questionnaires can be seen in Figure 7.4.

Sentinel IDS 2005
File Setup Analyze Operations

Setup Users Security Reports Sensors Mobile

User Security Questionnaire - Network Template

Please answer all the questions, this is so that Sentinel can be optimally setup. If you are unsure of what to select for a particular question, please select the most secure answer.

New Template
Name:

Update Template
Update:

Question 1 of 6

Vertical Port Scan Sensor:
The Sensitivity of this sensor should be set so that the 'Low' Value is the Maximum number of ports opened by a single host, before the sensor starts logging this as an attack. You should also set the Organizational Importance Level to allow the Sensor to know how strict you are on port scan activities! The sensor Timing is where you can set how often the sensor scans its tables for Attacks.

Click next to continue.

Set Template (Fuzzy Logic) High - Low Values:

Set Importance of this to your organization:
☐ Not Important
 ☐ Some Importance
 ☒ Quite Important
 ☐ Very Important

Set the Sensor Timing in Minutes:

Figure 7.4: Example of a Fuzzy Sensor Questionnaire

Once the sensors have all been initialized through the various systems' wired and wireless sensor template questionnaires, the resultant template will be saved in xml format, as stated previously. This template is the template graph, described in Chapter 6. The xml version of the templates will be transmitted to all the Sentinel sensors around the network. Figure 7.5 is an example of a template resulting from a questionnaire.

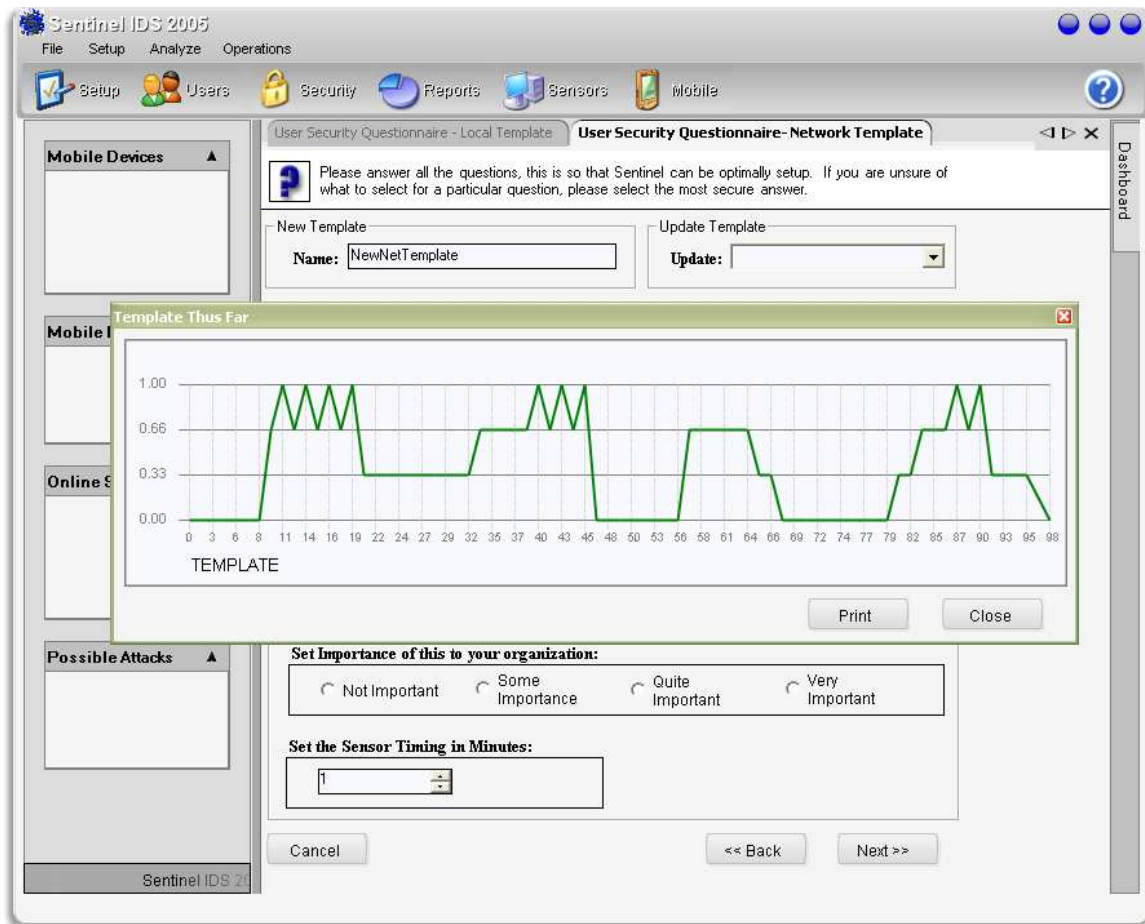


Figure 7.5: Example of a Fuzzy Sensor Template Graph

The next section explains the implementation of the neural engine component of the NeGPAIM-W prototype.

7.2.5 Implementation of the Neural Engine

As concluded in the previous section, this section focuses on the implementation of the neural engine component of Sentinel IDS. As stated in Chapter 6, this engine implements the anomaly-detection approach to intrusion detection, allowing both new and novel attacks to be detected. The engine does this by determining whether or not a user is performing normal or abnormal behaviour on the system.

The neural engine is implemented through neural networks, as it was concluded in Chapter 6 that this would be the best method for implementation. The neural networks have been implemented using an off-the-shelf, third-party tool, named Q-Net. This

commercial neural-network software package allows for easier implementation of the neural engine, while it is still in prototype. The neural engine's implementation is described in two phases: firstly, the collection of training data, and secondly, the training phase.

The first phase, as mentioned, is the gathering of data used to train the neural network. This is historical data, gathered randomly from the system, to prevent any bias. This data is gathered utilizing several metrics that have been identified. These metrics allow for specific data to be collected, providing data that is directly relevant to a user's activities on both the node and network on which he/she is working. Some examples of these metrics are listed below:

- CPU utilization;
- Network utilization;
- Frequency of network access;
- Network applications in use;
- RAM utilization;
- Applications in use; and
- Services in use.

This system data is then processed by Sentinel into a format usable by the neural network and stored in a centralized database in a total user behaviour string, described in Section 6.3.2. This total user behaviour string contains all the data that the neural engine uses for training. The string contains fields such as system behaviour pattern, network behaviour pattern, user behaviour pattern, user needs pattern and the relationship pattern which ties all the individual behaviour patterns together, and an example can be seen in Table 7.1. This example shows a user utilizing internet explorer for 170 seconds at an average network utilization of 25Kb/s. This would still be within his/her normal usage range, as the user normally uses internet explorer on average for 500 seconds, as can be seen by the user needs pattern field.

User Behaviour Pattern	System Behaviour Pattern	Network Behaviour Pattern	User Needs Pattern	Relationship Pattern
User Command	CPU Time Utilized (s)	Bandwidth Utilized (Kb/s)	Time Elapse (s)	Time Interval spent on Network Commands (s)
InternetExplorer.exe	170	25	500	20

Table 7.1: Total User Behaviour String

The second phase is the training phase, which takes the data collected and processed in the previous step and feeds it into the neural network. The neural network trains with this data and results in a profile of the user's usual behaviour pattern, based on the historical system data. Once this training has occurred, the neural engine is ready for deployment on the system itself.

The neural engine is implemented slightly differently to the fuzzy engine in that its main components are implemented on the main sentinel server, although it receives data from the sensors, as does the fuzzy engine. This data is not be processed on the sensor itself by the information refiner, but is rather forwarded to the main Sentinel server for processing. This saves resources on the sensors (as stated before, the hardware is very minimal on the sensors, and the neural network application utilizes a lot of resources). The sensors will, however, do the processing of input data, as mentioned previously, converting it into a format usable by the neural network.

The next section discusses the configuration of the neural engine, as has been explained in this section, with the use of screenshots from the actual Sentinel IDS.

7.2.6 Neural Engine Configuration

As was mentioned in the previous section, this section explains, with the aid of screenshots, the configuration process associated with the neural engine. This process was explained in high-level in the previous section, and this section focuses on the lower-level implementation itself.

The first step in the configuration of the neural engine is the selection of applications and processes that the user is allowed to run on the host and access network resources. Once these are selected, they are added to the legal processes list. The same is done for system applications and processes, where processes, which are to be allowed on the system and have access to network resources, are selected and added to the legal system processes list. These tasks can be seen in Figure 7.6, showing the initialization before training.

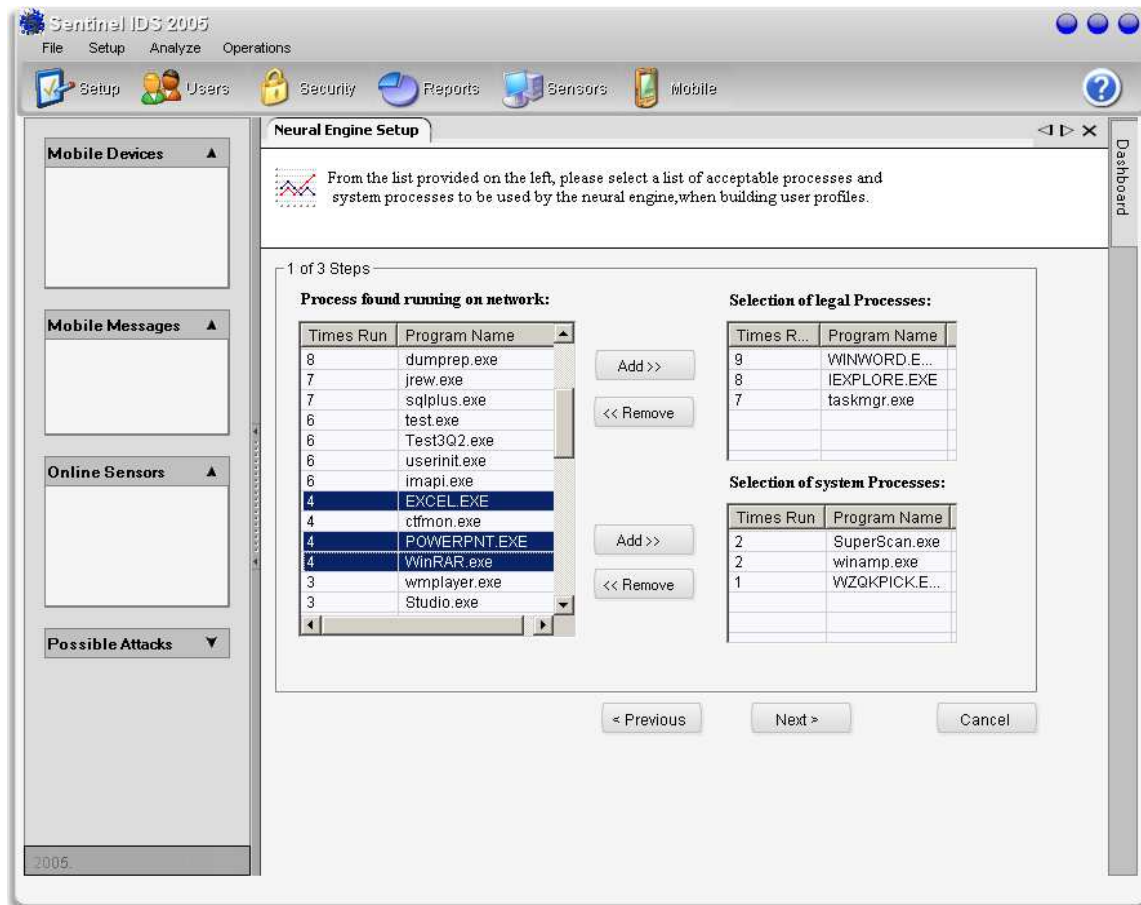


Figure 7.6: Neural Network Training (Initialization)

Once the initial selection of legal user and system processes is complete, the training of each user's neural network may begin. This process begins by the IDS determining the user's footprint from the data collected on the sensors, using the previously mentioned metrics.

After this process has completed, the neural network uses this footprint in the training process for the user. This process could take quite some time, depending on how many

users need to be trained for the system. An example of the training process can be seen in Figure 7.7.

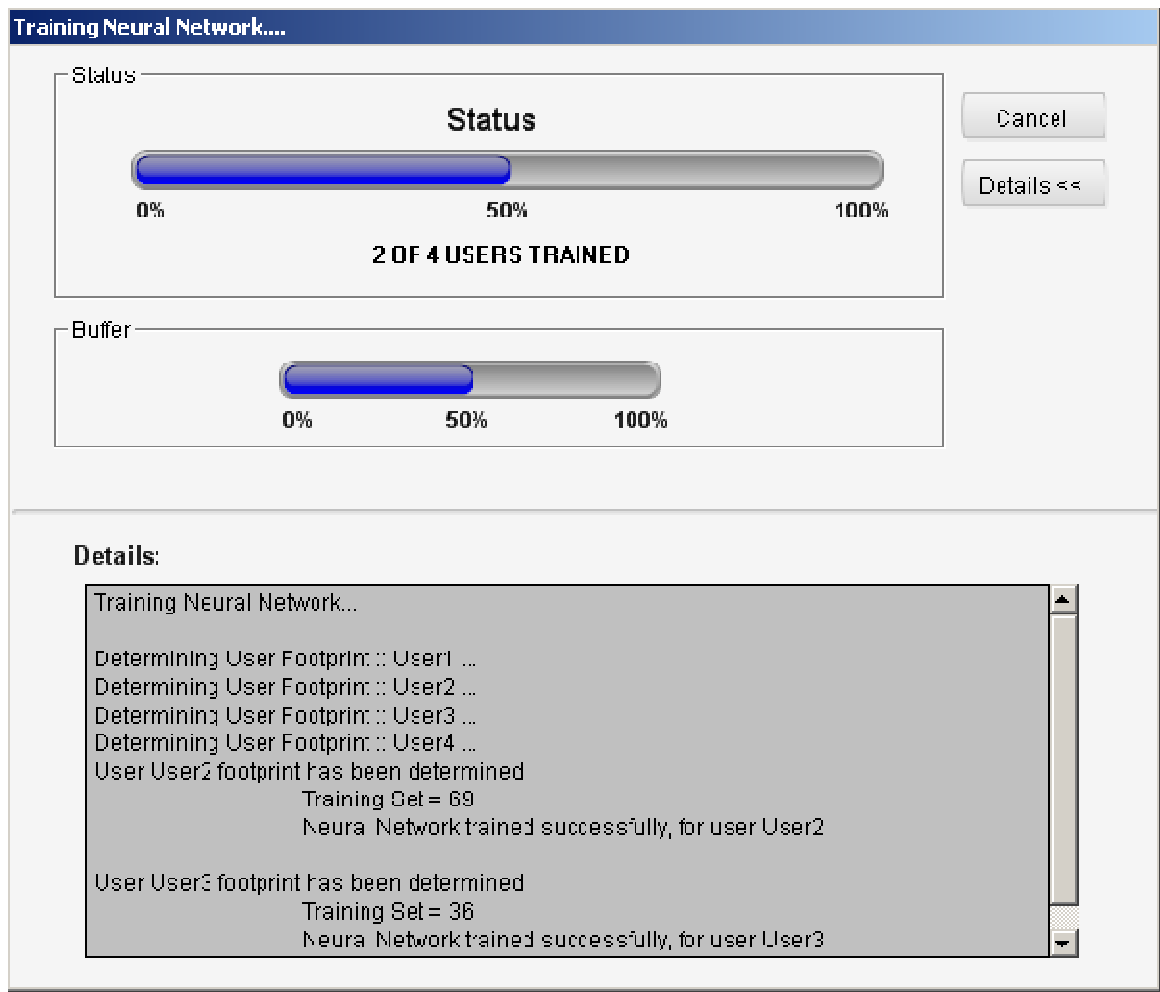


Figure 7.7: *Neural Network Training (Footprinting)*

Finally, when the training of the neural network is complete for all users on the network, Sentinel IDS will output information about the training process. This information includes the username, the size of the training set and the date of the last training event for the specific user. An example of this can be seen in Figure 7.8.

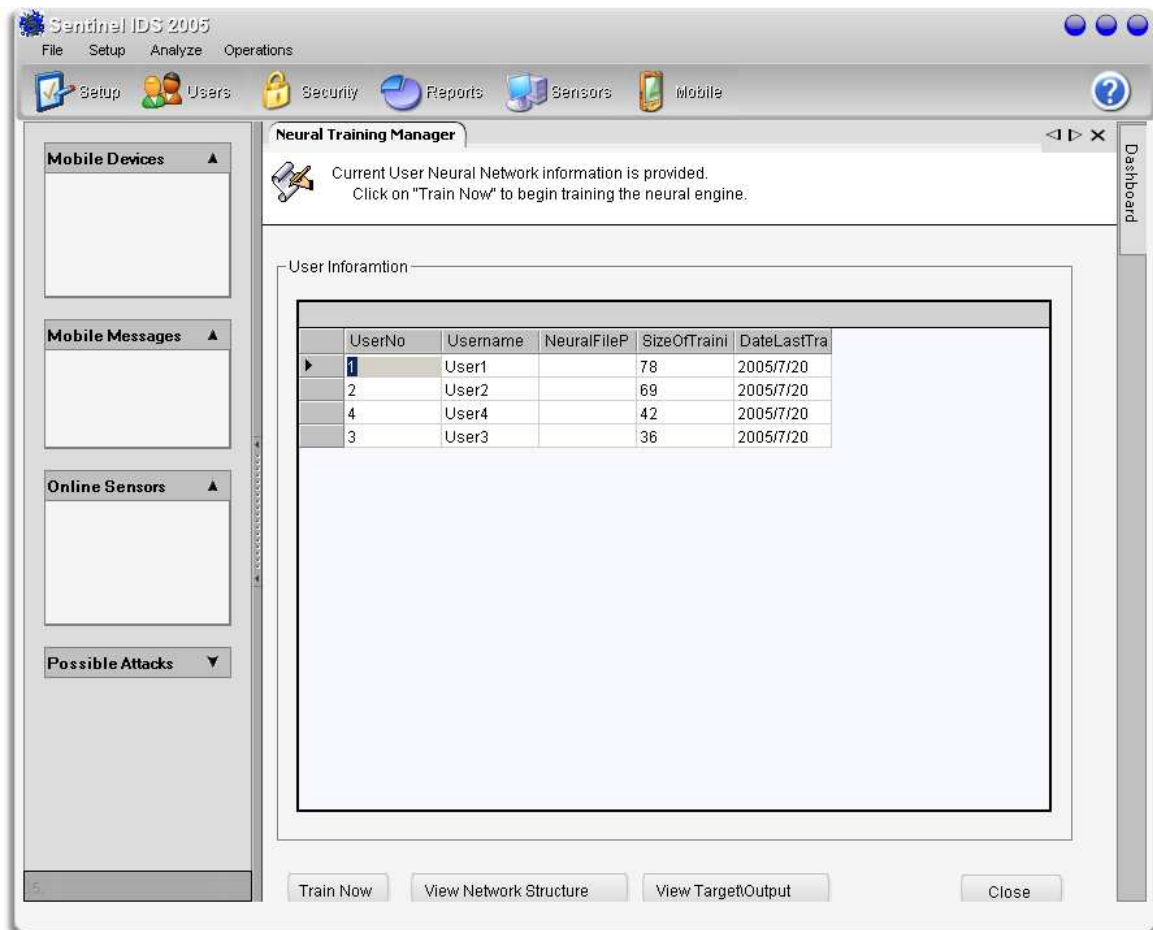


Figure 7.8: Neural Network Training (Completed)

In this and the previous two sections, one has seen the implementation and setup of the two low-level detection engines. The next section describes the single high-level engine. This engine provides the final part of the detection process, due to the fact that it determines finally whether or not an attack has taken place and if so, which responses need to be implemented.

7.2.7 Implementation of the Central Analysis Engine

The previous four sections explained the implementation of the two low-level detection engines. This section explains the implementation of the only high-level engine, known as the central analysis engine (CAE). As explained in Chapter 6, this engine is not a detection engine, but rather processes the output of the two low-level detection engines.

The CAE has been coded in such a way that the statistical calculations providing the engine's functionality have been hard coded into the software, and thus, cannot be changed or corrupted. The CAE, as mentioned in Chapter 6, takes in the outputs of both the neural and fuzzy engines as its input. The input data, taken from the two low-level engines, is then used by the CAE to determine the total possibility of attack, at which point, the engine implements various responses to the intrusion event.

The CAE implements these responses, based on how severe the attack is, as well as which network medium the intrusion occurred on. The responses to any particular attack can be either active or passive, explained in Chapter 6. Active responses are usually reserved for cases where intrusion attack or intrusive behaviour has been diagnosed as 50% certainty or greater. The reason for this is that active responses have an effect on network segments and user accounts. Therefore, passive responses are utilized to warn administrators to the possibility of attack, if the certainty is below 50%, and are implemented with active responses to alert the administrator to the actions taken when an active response is implemented.

The next section introduces the experiments performed on the Sentinel IDS implementation of the NEGPAIM-W Model. The environment, system and attack software is explained, so that one can better understand the experiments and the environment in which they were performed.

7.3 Prelude to the Experiment

This section discusses the software, intrusion tools and the actual physical environment in which the Sentinel IDS was tested. The experiments are explained in detail later in this chapter, and are set out as a case study, with the focus on the Sentinel IDS engines and the results gained during testing.

7.3.1 Experiment Environment

As mentioned earlier, the environment on which the NeGPAIM-W prototype Sentinel IDS is based is the Microsoft Windows family of client/server operating systems. The environment in which the experiment occurred was also, therefore, primarily based on the Microsoft platform, with most of the attacks running on a Linux-based node. This simulated a real-life environment, as most of the better wireless hacking tools are built for Linux (Lesser, 2001).

The version of Linux used in the experiments will be the Backtrack security auditing distribution. This distribution comes preloaded with all the intrusion and auditing tools needed to make the experiments successful.

The experiments were run in a laboratory environment at the Nelson Mandela Metropolitan University's (NMMU) Centre for Information Security Studies. The experiments were checked and overseen by two departmental employees. This was to ensure unbiased results on the experiments, including reports on the prototype. All results gained from the experiments are also reproducible with the Sentinel IDS software.

7.3.2 Intrusion Tools

As mentioned at the conclusion of the last section, this section contains a list and explanation of the intrusion tools that were used to perform the experiment. These tools were chosen specifically because they are all free and freely available for download on the Internet. Most of the intrusion tools to be used are Linux based. For most of the tools, there is a detailed explanation and tutorials on the Internet allowing potential attackers to perform intrusion attacks, even with little or no skills. Below is a list and explanation of the intrusion tools that were used to perform the experiments with the Sentinel IDS fuzzy engine.

1. **Airodump** (IronGeek, 2007) – is a Linux-based wireless tool that allows one to scan or sniff wireless access-points. Airodump will output to the console terminal a list of all access-points found and their MAC addresses, Channel, SSIDs and whether or not they have WEP encryption enabled. Airodump is

also used to dump wireless traffic from a selected access-point down to a hard disk. This data is used by other applications to crack WEP keys.

2. **Aireplay** (IronGeek, 2007) – is also a Linux-based software tool that allows one to take a wireless authentication packet collected off the network and replay the packet to the access-point. This process accelerates the data collection process, as many thousands of initialization vectors are needed in order to crack a WEP key.
3. **Aircrack** (IronGeek, 2007) – is an application able to crack a WEP key when it is passed enough data with initialization vectors. This data is usually collected by an application such as Airdump. Aircrack outputs the SSID, WEP key and MAC address of an access-point.
4. **SuperScan** (Foundstone Inc., 2007) – is a port scan tool, allowing an intruder to scan a host on a network and gain information as to which services are possibly running. This information is gained by checking which ports are open and relating that information to the services that run on the specific open ports.
5. **Brutus** (Hoobie Inc., 2007) – is a password brute forcing application, allowing an intruder to force his way into a system by brute forcing a password of a known user account. Brutus allows one to brute force through many kinds of protocols, including ftp, http, pop3, NetBIOS and many others.

This section has given the information needed as a background to the experiment, allowing one to see the tools that were used to perform the attacks, as well as the environment in which the experiment took place. The next section details the actual experiment. This takes the form of a case study. The case study is meant to serve as a proof of concept for the Sentinel IDS prototype, specifically focusing on the wireless detection components of each of the three main engines.

7.4 The Experiment

The experiments reported on in this section serve only as proof of concept with regards to the integration of wireless detection into the NeGPAIM Model. Only a select few of the countless possible wireless fuzzy sensors have been implemented in the prototype. The intrusion attacks chosen to be run against the system are some of the more common wireless attacks and show the newly updated fuzzy, neural and CAE engine's ability to detect wireless attacks. An explanation of the intrusion attacks performed, allowing the testing of the IDS, is as follows:

The Intrusion Attacks

This section begins the experiment by showing the intrusion attacks that have been set against the system. These attacks are widely used by real-world intruders to infiltrate and, in many cases, successfully steal and/or corrupt valuable organizational information. The attacks used to perform the experiments against the Sentinel IDS occurred as follows:

As mentioned previously, the environment in which this experiment occurs is a Microsoft Windows client/server environment, with two windows-based hosts that will be known as *Host1* and *Host2* for the experiment. *Host1* will be the host installed with the Sentinel IDS wireless sensors and is also the DHCP and FTP server. *Host2* will be the host housing the main Sentinel IDS application. *API* will be the wireless access-point with which *Host1* and *Host2* communicate. *Intruder1* will be the malicious intruder running a Linux-based notebook, who wishes to steal information from *Host1*. Now that the background has been given, see Figure 7.9, the next step is an explanation of the experiment.

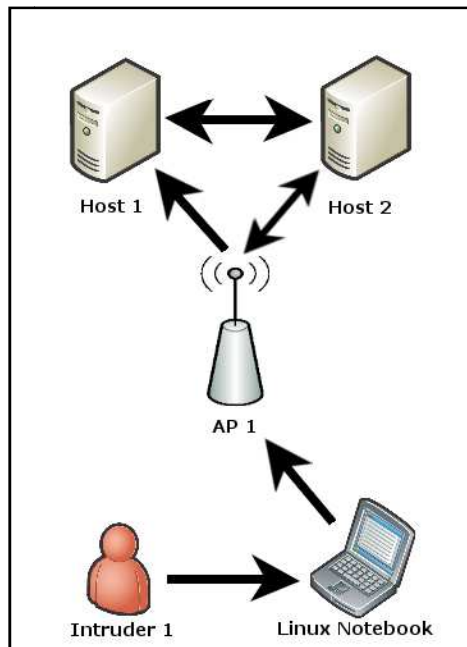


Figure 7.9: Setup of the Experiments.

Step 1:

First of all, the wireless card of the Linux-based notebook *Intruder1* was placed into monitor mode, allowing it to sniff all wireless traffic within its antenna's range. The network *AP1* was detected while sniffing for wireless signals. Now that *Intruder1* has the access-point's SSID, the next goal is to gain access to the wireless network itself. This sniffing of wireless signal can be seen in Figure 7.10, in which one can see that there are two access-points with WEP and one open. One can also see the various access-points MAC addresses and the number of packets transmitted all needed in later steps.

Shell - Console

Shell

CH 5][Elapsed: 2 mins][2006-11-26 15:23

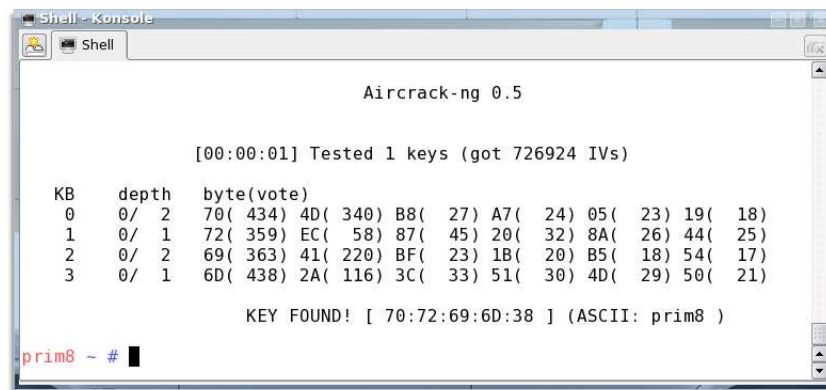
BSSID	PWR	Beacons	# Data	CH	MB	ENC	ESSID
00:11:95:9D:6C:84	-1	108	0	6	54	WEP?	linen
00:02:6F:42:BD:B4	-1	0	1173	2	-1	OPN	
00:02:A5:6E:E6:4A	-1	180	0	1	11	WEP?	

BSSID	STATION	PWR	Packets	Probes
(not associated)	00:13:02:06:40:30	-1	33	
00:02:6F:42:BD:B4	00:02:6F:05:C1:41	-1	1173	

Figure 7.10: Example Sniffing of Wireless Traffic.

Step 2:

To gain access to the wireless network, *Intruder1* had to collect wireless traffic destined for the *API*'s SSID. Enough data needed to be collected so that the WEP key could be cracked. To speed up the data collection process, *Intruder1* used Aireplay, which allowed him to replay initialization vector (IV) packets collected. Once the data had been collected, *Intruder1* runs Aircrack and gains the WEP key, which was "prim8". The cracking of *API*'s SSID is depicted in Figure 7.11, in which one can see the number of IVs collected and used to detect the key.



```
Shell - Konsole
Shell

Aircrack-ng 0.5

[00:00:01] Tested 1 keys (got 726924 IVs)

KB    depth  byte(vote)
0     0/ 2    70( 434) 4D( 340) B8( 27) A7( 24) 05( 23) 19( 18)
1     0/ 1    72( 359) EC( 58) 87( 45) 20( 32) 8A( 26) 44( 25)
2     0/ 2    69( 363) 41( 220) BF( 23) 1B( 20) B5( 18) 54( 17)
3     0/ 1    6D( 438) 2A( 116) 3C( 33) 51( 30) 4D( 29) 50( 21)

KEY FOUND! [ 70:72:69:6D:38 ] (ASCII: prim8 )

prim8 ~ #
```

Figure 7.11: Example Cracking *API* WEP Key.

Step 3:

After *Intruder1* cracked the WEP key, he/she then proceeded to connect to *API* with the newly acquired SSID and WEP Key. This allowed him/her initial access to the organizational network. The first action *Intruder1* took once he/she gained access to the organizational network was to perform a portscan with SuperScan on the network server *Host1*, which was located because it was the DHCP server, and assigned *Intruder1* an IP address when he/she connected. During the portscan of *Host1*, the intruder determined that *Host1* has an FTP server running on it. An example of this can be seen in Figure 7.12, in which one can see ports 21, 67 and 83 are open, these ports represent the FTP, DHCP and DNS protocols. This is how the attacker was able to gain information that an FTP service was running.

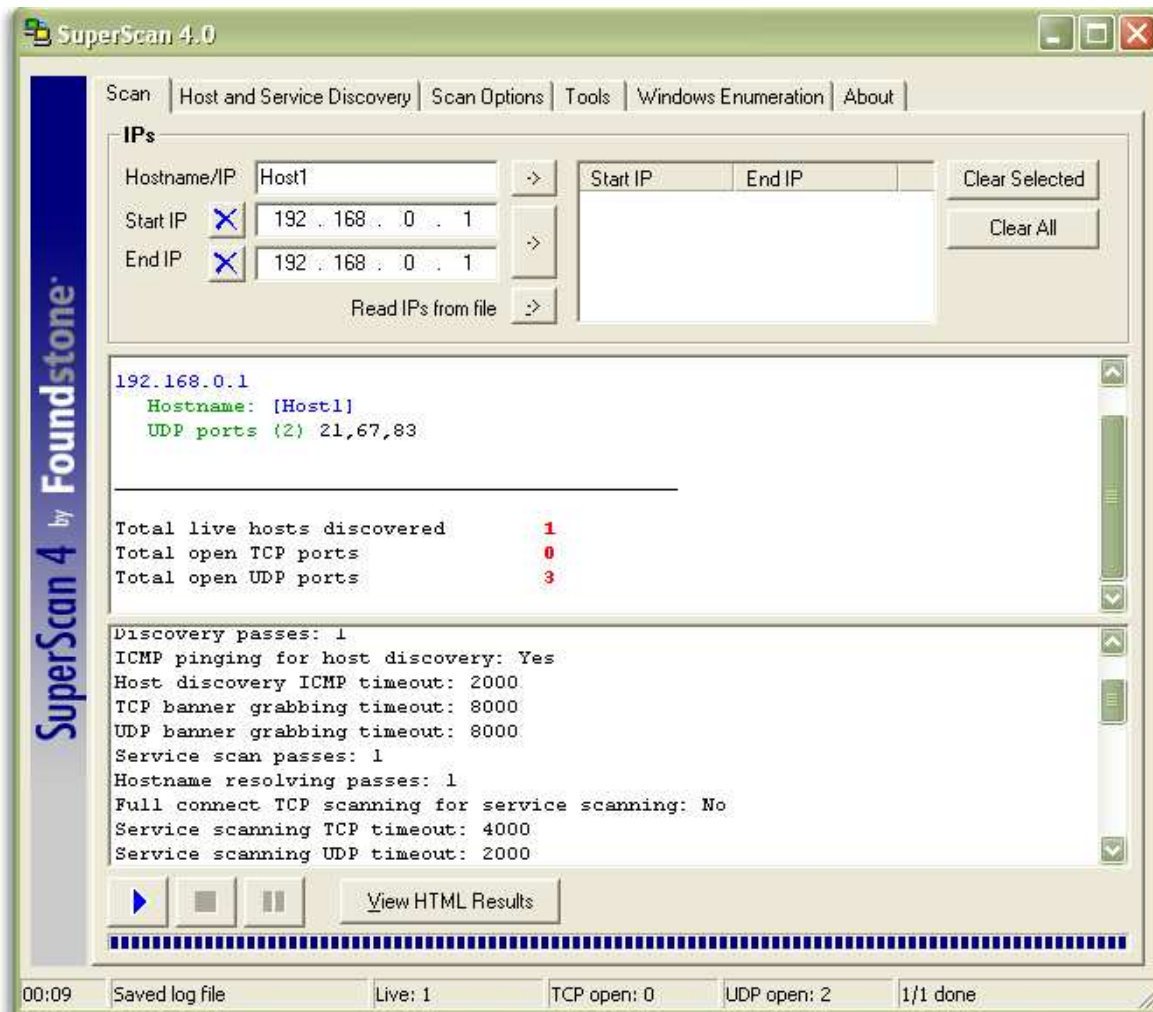


Figure 7.12: Example Portscan on Host1.

Step 4:

The final part of the attack occurred as *Intruder1* attempted to brute force the administrator accounts password on *Host1* using Brutus. The reason the intruder was unable to complete his/her attack and gain the password, which he/she was attempting to bruteforce, is due to Sentinel IDS sensors on both *Host1* and *Host2* determining that intrusive behaviour had taken place. This can be seen in Figure 7.13, in which one can see the target as *Host1*, the port as 21 learnt previously and the bruteforce type specified as FTP protocol. The user ID specified was Administrator; this was so that the wordlist would only run for one username, saving time.

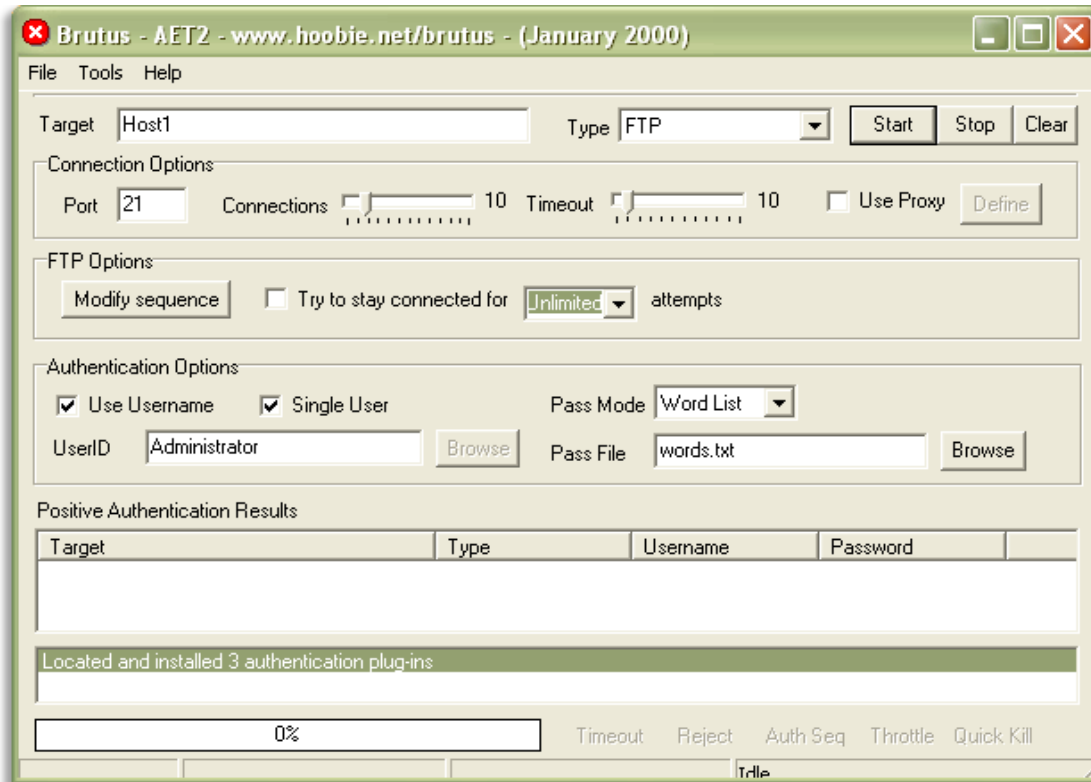


Figure 7.13: Example Bruteforce Attack on Host1.

The next section explains the results of the above listed attack as they relate to the three engines. As the output of the two low-level engines is needed for the high-level engine to calculate total attack probability (TPV), the two low-level engine results are shown and explained first.

7.5 The Results

After performing the experiment, the next logical step is to analyze and interpret the results. Thus, the main objective of this section is to report on the results of the experiments performed to test the Sentinel IDS implementation of the NEGPAIM-W Model. The results of the experiment are reported according to the individual engines, first the two low-level engines and finally the single high-level engine. The next section commences the result report with the first of the low-level engines, the fuzzy engine.

7.5.1 The Fuzzy Engine's Response

The fuzzy engine's response and output to the various parts of the attack mentioned in Section 7.4 is explained in this section. The fuzzy engine's response to the attacks is explained by the use of the specific fuzzy generic wireless or wired signature fired as a result of the intrusion. The results are presented as they pertain to the various attack steps described in Section 7.4.

Response to Attack Step 1:

In Step 1 of the Theft of Information attack, one saw the intruder placing his Linux-based notebook's wireless card into monitor mode (promiscuous mode). Within a minute or so, one of the Sentinel IDS fuzzy engine's wireless generic signatures fired. The signature that fired is one of the probing phase signatures, more specifically a signature which determined that *Intruder1*'s MAC address was not listed as a MAC address belonging to the organizations' hardware.

This in itself is not seen as too great a threat, as many other organizations in the vicinity may be utilizing wireless networks, but the fact that *Intruder1*'s wireless card is in monitor mode suggests that the owner may be sniffing for access-points. The fuzzy engine noted the MAC address of *Intruder1* (00:12:F0:3B:5F:71). This MAC address is kept for a predetermined time period for correlation purposes. The signature also started a timer to determine the length of time spent in monitor mode by *Intruder1*. The output of the fuzzy engine, at this point, is set at 5% probability of attack, as discussed in Section 6.2.2.2. This information was passed by the fuzzy engine to the CAE.

Response to Attack Step 2:

In Step 2 of the attack, one saw *Intuder1* detecting *API* as a viable target and utilizing Airodump to collect as much traffic containing initialization vectors as possible.

While *Intruder1* was collecting the data he/she needed to perform the next phase of the attack, two more probing-phase generic wireless signatures fired. The first signature fired as a result of *Intruder1*'s wireless card being in monitor mode for a period greater

than the limits allowed by the administrator. *Intruder1* spent a total of 3 hours and 5 minutes in monitor mode.

The second wireless generic signature fired during this step of *Intruder1*'s attack, fired as a result of unauthorized probing of *API* for its SSID. The fuzzy engine was set to report this as an act of misuse as all organizational wireless hardware have their MAC addresses registered and the probing was from a non organizational MAC. The output of the fuzzy engine at this point was set at 15% probability of attack as discussed in Section 6.2.2.2; this was due to the two fired generic signatures. The output was once again forwarded to the CAE.

Response to Attack Step 3:

At Step 3 of his/her attack, *Intruder1* had cracked the WEP key for *API*, from the data gathered during Step 2. The intruder then proceeded to portscan *Host1*, the DHCP and FTP server, which fired off another two wireless generic signatures.

The first of the signatures fired was an initial-access phase signature, fired during *Intruder1*'s first connection to *API*, using the SSID and WEP key. This signature had been monitoring *Intruder1*'s MAC address for an initial access, due to his/her prior data sniffing and probing activity.

The second signature fired was another probing phase signature, fired due to *Intruder1*'s portscan attack. The signature determined that the number and speed at which *Host1*'s ports were being accessed were past the limits set by the administrator.

The fuzzy engine's output at this point was set at 40% probability of attack as was discussed in Section 6.2.2.2. This information was then reported to the CAE as was done before.

Response to Attack Step 4:

As was mentioned in Section 7.4, the final part of *Intruder1*'s attack that he/she was able to commit was Step 4 in which *Intruder1* attempted to bruteforce *Host1*'s administrative account for its FTP server.

A hacking-phase system generic signature fired, due to the number of failed login attempts performed by the administrative account. The number of failed login attempts was set in the IDS on setup of the fuzzy engine. At this point, the fuzzy engine's output due to this and previous signatures fired was set at 75% as discussed in Section 6.2.2.2, and this output once again was forwarded to the CAE.

This concludes the fuzzy engine's output and response to the attack set out in Section 7.4. The next section outlines the neural engine's response to the attacks.

7.5.2 The Neural Engine's Response

As concluded previously, this section outlines and explains the various neural engine outputs and response to the intrusion attacks detailed in Section 7.4. As with the previous section on the fuzzy engine's response, this section explains the neural engine's response to the attack Step 4.

Response to Attack Step 4:

As mentioned previously, Step 4 is the point in *Intruder1's* attack in which he/she attempts to bruteforce *Host1's* FTP server using the administrator account. The neural engine, at this point, plays a role in the detection process. This occurs for two reasons.

The first is that the administrative account has had multiple failed login attempts and from the administrator's footprint, the neural engine determines that this is a deviation from his normal behaviour as he mostly enters his/her password incorrectly with the first attempt.

The second reason the neural engine determined anomalous behaviour for the administrative account is that the invalid password attempts took place during a period of time which is well out of the administrator's usual working hours which were 8am to 5pm.

The final piece of anomalous activity detected was due to the fact that the administrator usually works on a network node with a specific MAC address on the wired network, and the failed logon attempts occurred from the wireless network from a host with a MAC

address of (00:12:F0:3B:5F:71). This deviated dramatically from the administrator's daily habits.

The neural engine's output at this point was set at 70% certainty of attack, and its output was forwarded to the CAE for further action.

The next section provides the central analysis engine's response to the various attacks that took place, and also explains the various active and passive responses fired.

7.5.3 The Central Analysis Engine's (CAE) Response

This section gives an explanation of the responses of the central analysis engine to the attack that took place. As mentioned in Chapter 6, the CAE is responsible for statistically combining the fuzzy and neural engine output. Thereafter, it is responsible for implementing either active, passive or both responses, based on its output.

Response to Attack Step 1:

At this step in the attack, the CAE had only received intrusion probability from the fuzzy engine and after performing the statistical calculations, the CAE output is 2.5% probability of attack as described in Section 6.4.2. This is too low to implement any active or passive responses.

Response to Attack Step 2:

At this point in the attack the CAE had still only received intrusion probability from the fuzzy engine and, after performing the statistical calculations, the CAE output is 7.5% probability of attack as described in Section 6.4.2. This is still too low to implement any active or passive responses.

Response to Attack Step 3:

At Step 3 in the attack, the CAE had still only received intrusion probability from the fuzzy engine, and after performing the statistical calculations, the CAE output is 20% probability of attack as described in Section 6.4.2. This output is enough to cause the CAE to implement a passive response, and the administrator of the network was e-mailed

regarding the results. The passive response was implemented due to the configuration choices of the administrator, when he/she set the threshold values for the CAE.

Response to Attack Step 4:

At this final step in the attack, the CAE had finally received intrusion probability from the fuzzy engine and the neural engine. After performing the statistical calculations, the CAE output is 72.5% $((70\% + 75\%) / 2)$ probability of attack as described in Section 6.4.2. This output is enough to cause the CAE to implement both active and passive responses. The passive response was to e-mail the administrator of the network. The active response was to block Intruder1's MAC address on all access-points, including *API*, and to disable the account administrator on *Host1*.

As can be seen from the abovementioned experiments, the wireless extensions to the original NeGPAIM Model most definitely provide protection against the wireless intrusions. These intrusion attacks, as previously mentioned, are the more common intrusion attacks, and easily performed by anyone with access to the attack tools. The experiments performed in this chapter are not the only attacks researched and tested during this research. These are only a few of those tested over the past two years to prove the validity of the research conducted.

7.6 Conclusion

This chapter has shown that the Sentinel IDS implementation prototype for the NEGPAIM-W Model indeed shows excellent initial results, with detection of both wired and wireless intrusion attacks. With the explanation of both low-level detection engines and the implementation of the single high-level engine, the Sentinel IDS allows for an excellent means to both detect and prevent wireless-based intrusion attacks.

Through the experiments performed, one can see both the validity and need for the NEGPAIM-W Model. The next chapter concludes the dissertation by focusing on topics, such as the objectives of dissertation, achievements and possible further research plans.

Chapter 8

The Conclusion

8.1 Introduction

Chapter 6, contained an explanation of all the main components of the NeGPAIM-W Model with the main focus placed on the two low-level engines and lastly on the single high-level engine. These engines together form the backbone of the NeGPAIM-W Model, allowing it the ability to detect both wired and wireless intrusion events and report on these intrusion attempts. In Chapter 6, the model and the main engines were explained theoretically, and in Chapter 7 the implementation of the model and the three main engines in the Sentinel IDS prototype were described.

The objective of Chapter 7 was to determine several points, the first of which was to prove that intrusion attacks are indeed a reality, and that one needs to protect one's organization from these intrusion attacks. The second determination that Chapter 7 attempted to make was that of testing the NeGPAIM-W Model's ability to detect wireless-based intrusion attacks. This was done via the use of the Sentinel IDS prototype. As was seen in the actual experiments, the Sentinel IDS prototype was able to detect the wireless-based attacks proactively throughout the intrusion, culminating in the halting of the intruder's progress in his/her attack.

The attacks used in the testing of the NeGPAIM-W prototype in Chapter 7 were all real-world intrusion and penetration testing tools. These, as was discussed previously, were used so that the results of the experiments would be as close to real-world conditions as possible. When comparing the results obtained from the experiments in Chapter 7 with research conducted by Mell et al. (2003), the results seem very favourable. Thus, the results of these experiments have been taken into consideration in the rest of this chapter while determining whether or not the

objectives stated in Chapter 1 have been met. The next section contains a review of the problem statement.

8.2 Review of the Problem Statement

This dissertation addressed the problems associated currently with wireless intrusion attacks in organizational environments. The research conducted also showed the need for IDSs to adequately protect an organization's information assets proactively from attacks, both internal and external to the organization and on any network media. These problems were addressed during this dissertation in the following way.

- Combating wireless intrusion attacks (Refer to Chapter 1.2):

Wireless intrusion attacks were discussed during this dissertation, and it was concluded that these wireless-based intrusions are just as lethal to the information assets of an organization as their wired counterparts. It was also determined that if the wireless networks were not adequately protected, the overall security of the organization can be compromised by an intruder.

For this reason, implementation of security mechanisms, such as intrusion detection systems, are critical within wireless networks. This is made even more evident when one looks at how easy it is to gain access to a wireless network from outside an organization, due to spillover from wireless access-points, and the poor security provided by wireless equivalent privacy (WEP).

The various security mechanisms were discussed in Section 2.4, with Sections 2.5 and 2.6 focusing on intrusion detection systems. The various wireless security problems were addressed in Section 3.4, with the conclusion being made that it is best to implement both OSI layer 2 and layer 3 security mechanisms in the protection of wireless networks. This, because the broadcast nature of wireless security allows an intruder to collect data off the wireless network even though there may be layer 3-based protection mechanisms.

- Proactive detection/prevention of wireless intrusion attacks (Refer to Chapter 1.2.1 and 1.2.3):

It was determined during this dissertation that attackers can have one of four goals when performing an attack against a network. These goals are the corruption of information, theft of information, denial of service and theft of service. These form the basis of the updated alternative approach to misuse detection.

For an intruder to achieve one of the aforementioned goals, he needs to perform certain actions. These actions can be tracked by categorizing them into one of the six generic intrusion phases. The weightings assigned to these generic phases differ greatly on a wireless network, as opposed to the same phases on a wired network. With the new weighting system in place, wireless attack detection is finally a reality, and in some cases, the actual payload of an attack may even be prevented.

The abovementioned problem was discussed in detail in Section 6.2.2. The proposed solution was, as mentioned above, to change the weightings of the six generic phases in the case of a wireless intrusion attack. This was done as wireless attacks place more emphasis on probing and initial-access phases; thus, the weightings need adjusting. With this in mind, the sooner a wireless attack can be detected, the less damage the intruder can cause over the wireless network. In some cases, correlation of wireless and wired attack data may even cause an alert to a greater threat than was previously known.

- Detection of existing, new and novel attacks (Refer to Chapter 1.2.2 and 1.2.4):

The NeGPAIM-W Model makes use of both misuse and anomaly detection to detect intrusion attacks; thus, the model has the ability to accurately detect known attacks through the use of misuse detection and new or mutant attacks due to its anomaly detection engine. An example of how the anomaly engine detects a new or mutant attack is as follows: the anomaly engine detects attacks based on the events leading up to and during the attack and the reaction of the system to the attack e.g. CPU usage, RAM usage and network usage. For instance, although a new or mutant DoS attack may work

slightly differently, it still attempts to deny service in some way to legitimate users. With this in mind one can see that no matter what type of DoS attack occurs, certain metrics such as excessive network or CPU utilization will be recognized by the anomaly engine as abnormal activity.

This is important as new attacks are released almost hourly, and pre-existing attacks are detected through the NeGPAIM-W generic wireless and wired network signatures; thus, the networks are protected on two fronts. The final and definitive determination is done by the central analysis engine which, as previously mentioned, does not do any actual detection, but performs statistical analysis on the anomaly and misuse detection engine output.

Chapter 6 as a whole discussed and focused primarily on these three engines, with an in-depth explanation of the inner workings of all three of the engines contained in Sections 6.2, 6.3 and 6.4 respectively. With the use of two forms of intrusion detection and the CAE's combination of the results of the low-level engines, proactive detection of old, new and novel attacks is possible, allowing systems protection against previously undetectable intrusion attacks.

8.3 Meeting the Dissertation Objectives

This section reviews the objectives of this dissertation as laid out in Chapter 1, Section 1.3, which presented three distinct objectives needed to be addressed through this dissertation. During the course of the research, many other objectives have also been met and is discussed later in the chapter. The following paragraphs evaluate to what extent this dissertation and the NeGPAIM-W Model has attained these primary objectives.

Primary Objective:

Investigation of Wireless Intrusion Attack Effects:

The first primary objective of this study was to investigate wireless-based intrusion attacks and the effects they have on currently available IDS products. Chapters 3 and 4

provided insight into the world of intrusion attacks, specifically focusing on wireless intrusion attacks.

Sections 4.4 listed and explained various commercial, research and public domain IDS products, which were compared head to head. This demonstrated the lack of wireless detection components within these IDSs.

In Section 4.5, the limitations of currently available IDSs were listed and explained, so proving the need for a new model, such as NeGPAIM-W; thus, meeting this objective.

Thus the main objective as set out in Section 1.3 has been satisfactorily met during this research, with the many secondary objectives also subsequently being met. These secondary objectives are briefly listed below.

Secondary Objectives:

- **Objective A:**

To determine what could be done to minimize the damage done by wireless intrusion attacks. This objective was achieved by Section 4.6 along with Chapters 5 and 6. This was achieved with Section 4.6 containing a discussion on the characteristics of a wireless IDS, with wireless IDSs being recognized as the best method for minimizing wireless intrusions. This objective was further met through Chapters 5 and 6 with the NeGPAIM-W model an example can be found in Section 6.5, showing how the NeGPAIM-W model limits an attack.

- **Objective B:**

To create a model that would be proactive in nature and have the ability to detect both wired and wireless attacks, the model is proactive due to its ability to identify the attacker's user ID as well as determining the attacker's next step. This objective was met through defining an updated NeGPAIM-W Model, detailed in Sections 6.2.4 and 6.3.

- **Objective C:**

To investigate computer crime, particularly focusing on proactively detecting wireless intrusion attacks. This was addressed primarily in Chapter 2.

- **Objective D:**

To update the previous NeGPAIM Model, thus allowing the model to cater for wireless intrusion attacks, including the correlation of wireless attacks with attacks taking place on wired network segments. This was addressed in Chapter 6.

- **Objective E:**

To create a prototype for the NeGPAIM-W Model, and perform an experiment on the prototype, utilizing real-world wireless intrusion tools. This has been addressed in Chapter 7.

8.4 Further Research

Although all the objectives set out in Section 1.3 have been met within this dissertation, there are certain components and concepts that could be further expanded. These are listed below:

- **Identification of further wireless sensor signatures:**

Although there were a few generic signatures identified during Chapter 7, there are countless other generic signatures that could be researched e.g. signatures for attacks on router and switch hardware, allowing the further detection of less common wireless attacks. The generic signatures that have been identified in this dissertation have been aimed at detection of the more prevalent wireless intrusion attack groups.

Further research could enable the NeGPAIM-W Model and its prototype Sentinel IDS to detect a greater number of intrusion attacks occurring, while, at the same time, increasing the accuracy of the IDS by allowing attacks to be sub-grouped. This occurs as follows: attacks are detected first by attack family e.g. DoS attack and then the attack is sub grouped into say wireless-based DoS attacks or wired-based DoS attacks etc.

- **Identification of further wireless metrics:**

As was discussed in Chapter 6, a few metrics have been identified for use with the training of the neural network. To improve the detection capabilities even further than they are currently, new suitable wireless, wired and system metrics need to be identified

e.g. network utilities / applications running, ports open for a specific user and the wired / wireless network usage hours for a specific user.

The greater the number of metrics available, the greater the knowledge of a user's actions will become. With few metrics, although accurate, only a subset of the user's actual effects on the network will be known. With further research, the detection capabilities of NeGPAIM-W could be far better.

- **Identification of further system responses:**

The system responses to both wired and wireless network intrusion events were briefly covered in Chapter 7, and these responses would need further research. This further research into the responses would allow the IDS to respond to different intrusive events, according to their severity, thereby limiting the cost to overall network usage. Currently, there have only been a few responses that have been identified.

- **Linux-based wireless sensor signatures:**

With the number of wireless devices running on the Linux platform, it would make sense to conduct further research by determining what signatures could be implemented on the Linux environment. This would also mean creating wireless sensor software for the Linux platform.

The outcome of this research would allow one to compare the resource requirements of each platform, when running the wireless-based sensors. This information could be critical to a commercial implementation.

8.5 Conclusion

Wireless networks are rapidly becoming part of the way organizations do business. With the inherent security risks associated with currently available wireless technologies, there is a need for good security mechanisms. Organizations that implement wireless technologies must take the required security mechanisms associated with wireless technologies to heart from the outset,

and not simply as an afterthought. Security mechanisms, such as NeGPAIM-W, allows organizations to keep their wireless networks safe.

NeGPAIM-W has been implemented, utilizing the latest technologies, in a prototype known as Sentinel IDS. This IDS allows for the proactive detection of both wired and wireless intrusion attacks. The experiments conducted on Sentinel IDS show that the updated model can protect an organization from attacks performed on both wired and wireless networks; therefore, the main objectives outlined in Section 1.3 have been fully met.

The updates to the original NeGPAIM Model allows it to keep modern networks safer for organizations, allowing organizations to be less worried about their security and concentrate more on their core business.

FINALLY, IT IS CLEAR FROM THE RESEARCH THAT NO ORGANIZATION CAN BE WITHOUT ADEQUATE SECURITY MECHANISMS AND THAT NeGPAIM-W WILL IMPROVE MOST, IF NOT ALL OF AN ORGANISATION'S INFORMATION AND COMPUTER SECURITY. THEREFORE, NEGPAIM-W, OR THE CONCEPT OF THE MODEL, HAS THE POTENTIAL TO BECOME THE IDEAL MODEL TO PROTECT AGAINST INTRUSION ATTACKS.

Annexure A

Utilizing Fuzzy Logic and Neural Networks for Effective, Preventative Intrusion Detection in a Wireless Environment

Robert Goss & Martin Botha

ABSTRACT: *The importance of properly securing an organizations information and computing resources has become paramount in modern business. Intrusion detection systems in particular have an increasingly valuable role to play, as networks grow and more information becomes available, administrators need better ways to monitor their systems. Most current intrusion detection systems lack the means to accurately monitor and report on wireless segments within the corporate network, this paper will propose an extension to the NeGPAIM model that will allow for the accurate detection of attacks originating on wireless network segments. This will be done by the use of Fuzzy logic and Neural networks utilized in the detection of intrusion attacks. The model is based on the assumption that each user has and leaves a unique footprint on a network when using it. This model is able to proactively detect intrusion attacks in both wired and wireless environments.*

KEYWORDS: *Computer security, intrusion detection, wireless intrusion detection, intrusion detection systems, intrusion attacks, wireless networks, NeGPAIM model, NeGPAIM-W².*

1. INTRODUCTION

Since the inception of computers, they have become an integral as well as an indispensable part of our everyday lives. Information Communications Technology (ICT) has been, and is currently advancing at a rapid pace (LT Consultants et al, 2002; Intel Corporation, 2005). Internet which is one of the main implementations of ICT is increasing rapidly at rates better than that of any other communications technology to date. In 2000 it was estimated that approximately one half of US households were online (Wilhelm, 2000). The number of Internet hosts online has also increased from 44 million in January 1999, to 88 million in August 2000, to almost 120 million in April 2001 (Telcordia Technologies, 2001). Web commerce has become mainstream, with millions of people buying online yearly. New consumers are now on the scene, buying items online instead of in a regular store, these consumers are known as cyber consumers (LT Consultants et al, 2002). With business being conducted over the Internet, businesses have had to make information available to individuals outside of their organization (DeYoung et al, 2002). This availability of information has lead to security holes also becoming available to the public facing web servers on the businesses network.

With fast wired network technologies such as gigabit Ethernet becoming increasingly prevalent as corporate network backbones, coupled with the current need for mobility. Many organizations are implementing wireless networks, one of the current buzzword technologies in ICT. With the implementation of wireless technologies, these organizations become vulnerable to a plethora or new intrusion attacks. The need for proper protection of a company's information is thus becoming more and more important every year, this can be seen in the Annual CSI/FBI Cyber Crime Survey, which estimates that the amount of money lost by companies due to attacks, system breaches and theft of information is around the figure \$141,496,560 this is down from 2003 when the total losses were estimated at \$201,797,340, (Gordon et al. 2005). There are many ways to protect data e.g. Firewalls, Antivirus, Access Control, Policies and Intrusion Detection Systems.

The purpose of this paper is to discuss, firstly, some of the many problems currently associated with IDSs and secondly one potential solution, in the form of a model known as NeGPAIM-W². The paper will commence with an introduction to the various types of IDS available to an organization along with the technologies problem areas. Following this is an introduction into the previously mentioned NeGPAIM-W² model. The paper culminates with an experiment on the model, including results, the paper concludes thereafter.

2. PROBLEMS WITH CURRENT IDSs

Intrusion detection has been around in some form for the past twenty six years, but has only taken off within the last ten or so years (Innella, 2001). Before the mainstream use of intrusion detection and intrusion detection systems, organizational networks would be hacked with no warning given. In an attempt to detect an intrusion, administrators would manually scan log files in an attempt to determine what happened to the network, the identity of the attacker and source of the attack. The problem was, that the process was a time consuming activity and more often than not resulted in no findings.

The alarming figures reported by the CSI/FBI survey shows that security on organizational information systems is not just an afterthought but should be one of the main concern's, when setting up or upgrading a network (Gordon et al. 2005). This is evident by the fact that ID and IDSs have undoubtedly become an indispensable protection mechanism to any organization (Li, Das, & Zhou, 2005). Today's

trend is moving towards intrusion prevention and proactive intrusion detection both of which attempt to limit or completely stop an attack dead in its tracks. By utilizing artificial intelligence and other techniques, IDSs are able to determine if an event is possibly part of an attack or not.

Intrusions are no longer limited to an attacker being physically attached to the network through some cabled medium. With wireless access becoming more prevalent and with many organizations investing in wireless technologies, without first understanding the dangers, many of these organizations have gaping holes in their security. Attackers can sit outside the office building of a target and connect directly into the network and attack it via the organizations own wireless infrastructure and then disappear. Most commercial IDSs available currently do not have wireless detection capabilities, this problem leaves organizations open for attacks. Therefore it is quite clear that there is now a need for a more holistic approach to ID than is currently available.

There are currently three main types of Intrusion Detection Systems namely HIDS (Host-Based IDS), NIDS (Network-Based IDS) and Hybrid IDS which combines the two afore mentioned types to form a more rounded IDS. HIDS's reside on a single host and usually monitor and protect the system configuration and files from abnormal changes, files and system settings are given weightings and an administrator can be alerted upon suspicious activity. In NIDS's the IDS will monitor multiple nodes on the network and detects attacks by searching network traffic for patterns of known attacks and anomalous activity known from previous baselines. This traffic could have a source external to the organization, but have an IP address of a machine internal to the network; this is known as IP spoofing and NIDS's can be setup to identify this kind of attack. Both NIDS and HIDS require a database of previous known attacks to detect most attacks (Lehmann, 2005; Whitman & Mattford, 2003). The problem with these two approaches is that there is little to no correlation between network-based and host-based intrusion events. This is mainly because intrusion information is stored in separate database.

There are two main methodologies of detection to which most IDS's subscribe; these are Misuse Detection and Anomaly Detection. Misuse Detection usually works by using a database of known attacks and compares current User actions to the database using rule based systems. This is also known as signature detection and functions much like an antivirus as a misuse detection system is only able to detect new attacks if you the administrator keep the signature database up to date. Another problem associated with misuse detection is that if an attacker performs his attack over a long period of time, the attack may not be picked up. The only way to currently combat this problem is to collect and analyze data over a large period of time.

Anomaly detection typically utilizes threshold monitoring to indicate when a certain established metric has been reached. Anomaly based IDS's are also known as behaviour-based IDS's because of the way they function. IDS's based on anomaly detection collect data from normal traffic or user actions and over time establish a baseline. When the system has a baseline the IDS will take samples of network traffic and compare it to the baseline using statistical calculations, if the collected data deviates to widely from the baseline; then the administrator is notified. The baseline is usually created using some of the following metrics: CPU usage, memory usage, network packet types, user typing rate etc. A problem associated with anomaly detection is that a user over time can train the system to accept anomalous behaviour as normal, by slowly adding to the attack (Whitman & Mattford, 2003).

With the many problems associated with modern IDSs, a new model for ID has been formulated. This model known as Next Generation Proactive Identification Model with Wired and Wireless (NeGPAIM-

W²), and is based on research performed previously at the Nelson Mandela Metropolitan University (NMMU). The next section will detail the NeGPAIM-W² model, including its primary components.

3. THE NeGPAIM-W² MODEL

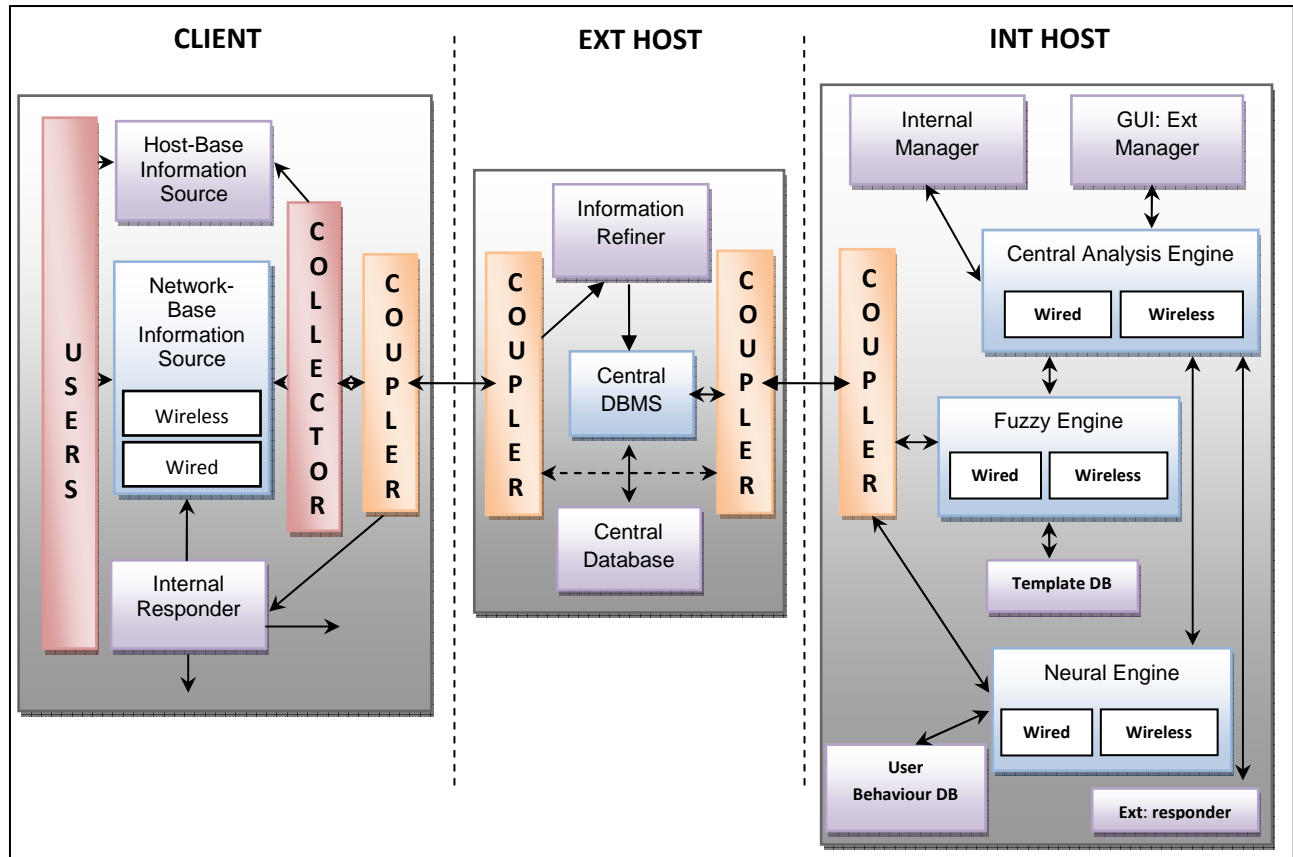


Figure 1: General Representation of NeGPAIM-W²

There are nine main components making up the NeGPAIM-W² model, these are as follows: Information Provider, Collector, Coupler, Information Refiner, Fuzzy Engine, Neural Engine, Central Analysis Engine (CAE), Responder and Manager. Three of these components are directly involved in the detection of intrusion attacks, namely the fuzzy, neural and CAE engines detection engines. The fuzzy and neural engines are known as the low-level detection engines and the CAE known as the high-level detection engine.

For the purposes of this paper, the focus will be primarily on the function and operation each of these three detection engines. Both low-level and high-level detection engines will be discussed below.

- **Fuzzy Engine:**

The fuzzy engine is one of the two low-level processing units of NeGPAIM-W² and will process the input data. This engine is responsible for implementing the Misuse Detection methodology. The fuzzy engine will compute a template firstly, and the user action graph will be mapped against it to determine whether or not a user (intruder) has been, or is performing an intrusion attack.

The overall intrusion probability for the network sensors is divided into two weighted parts: one weighting for the wireless attack probability and another weighting for the wired network intrusion attack probability. The fuzzy engine's network detection rules have been updated with the new NeGPAIM-W² Model to provide a better detection rate. The rules have been updated to detect attacks at layers 2 and 3 of the OSI model where the previous fuzzy engine specifically targeted layer 3 only. The updated model allows for better performance by separating the network detection into wired and wireless separately, and weighting the outputs to form a final fuzzy intrusion attack probability as seen in Figure 2. This also allows the engine to take into account the differences in transmission of data over the different network mediums.

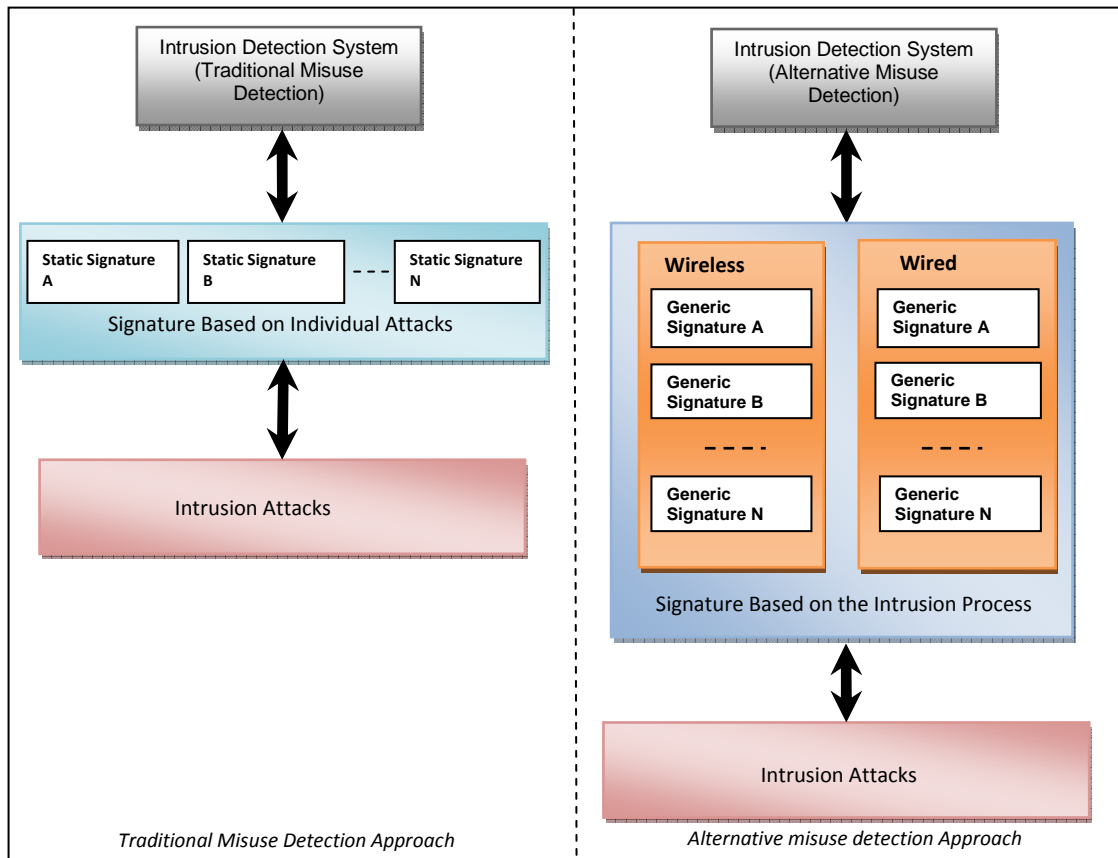


Figure 2: Comparison between Traditional and Alternative Misuse Detection

The fuzzy engine will pass its intrusion probability value to the central analysis engine. This is a continuous process.

- **Neural Engine:**

The neural engine is the second of the two low-level processing units and will also process input data. This engine will process the data and search through it for patterns of abnormal user behaviour that may be occurring.

This abnormal user behaviour may come in through one of three sources: the host-based sensor, the application-based sensor or the network sensor. The network-based sensor is what this section will explain. The neural engine uses a user's wireless and wired network usage

patterns to determine whether or not the user is acting abnormally on the system. For instance, the user may work via a wired terminal from 8am to 5pm, Monday to Friday. Then one day he/she logs into the network on a Sunday afternoon over a wireless connection. This will be noticed by the neural engine as anomalous activity.

The engine reports abnormal user behaviour to the central analysis engine by way of intrusion probability value.

- **Central Analysis Engine (CAE):**

This is a high-level processing unit, the objective of which is not to perform anomaly or misuse detection, but rather to analyze and interpret the resultant output values from the fuzzy and neural engines as well as managing the other units of the model.

The CAE performs four critical operations for the categorization and halting of attacks, these operations are as follows:

1. To determine the source of an attack, be it wired, wireless or both network media. This information is utilized to determine what if any action will be taken against the attacker.
2. To determine the type of attack being currently perpetrated by the attacker, this is needed to better understand what actions he/she may take next.
3. To take into account all information gathered from various sources and to determine an overall intrusion probability.
4. Finally the engine uses the overall intrusion probability value along with the type of, and source of the intrusion attack to perform a response to the intruders actions.

The engine outputs to the administration, the final intrusion probability with final weighted scores of attack type, source, etc. This is generated after performing the statistical calculations on the output of the two lower-level units. Figure 3, Depicts the interaction between the different components of the NeGPAIM-W² model, with the focus on the three detection engines.

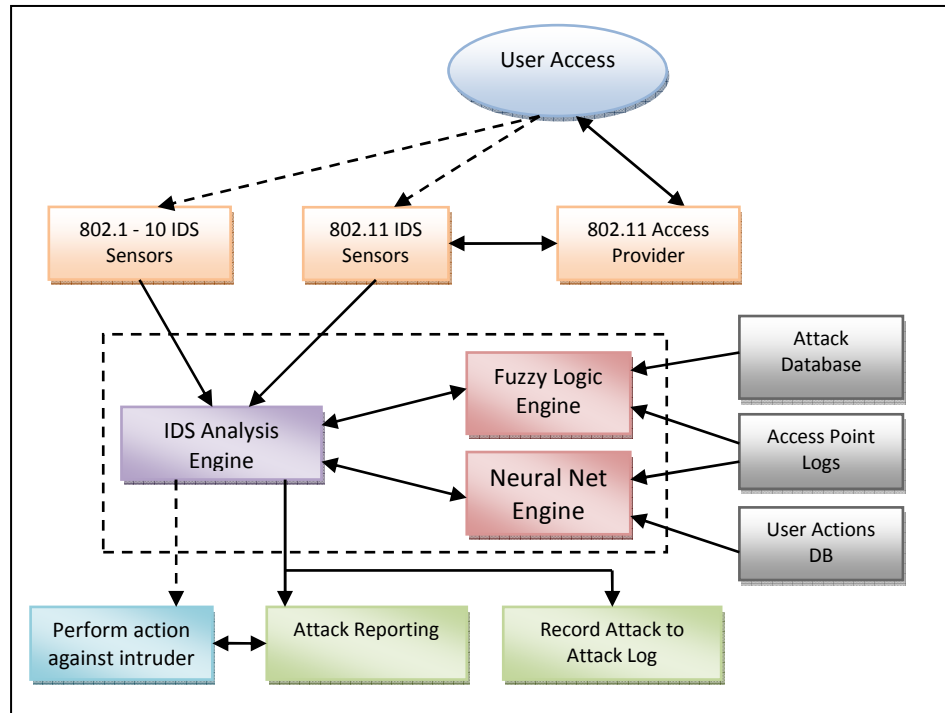


Figure 3: Low Level Detection Model

The nine major components of the updated model seen in Figure 1, are divided into a three-tier hierarchical hybrid architecture. The three tiers are as follows: Client, External Host and Internal Host. This architecture allows for benefits of performance and security to the NeGPAIM-W² Model and will allow the security administrator the ability to monitor his/her network for attack more efficiently. This is important because, as wireless networks become more sort after, widespread and faster, they will become the main way for users to connect to organizational networks. NeGPAIM-W² is a proactive next generation IDS, which not only caters for wired networks, but also has the ability to monitor wireless networks. NeGPAIM-W² is a theoretical model, and for any model to be accepted, it must be tested practically. The next section will show the results of one of the experiments performed on the NeGPAIM-W² prototype.

4. THE EXPERIMENT

Key elements of the NeGPAIM-W² model have been implemented in a fully functional prototype named Sentinel IDS. These elements are namely the reporting, Fuzzy, Neural and Central Analysis Engines; the reason these elements were chosen being that they form the core backbone of detection and the feedback processes allowing for the proactive detection of attacks. Responses, both passive and active have been implemented as well as remote sensors (smart agents). The Microsoft Windows environment was chosen as the test bed for the NeGPAIM-W² Model, although the model could be adapted to fit the Linux/Unix environments with little trouble. It was decided to utilize MS Windows, because it is harder to gain access to system statistics and to monitor network activity in Windows as opposed to Linux/Unix. This experiment will therefore not only prove that NeGPAIM-W² is feasible, but that it can be implemented in many different environments.

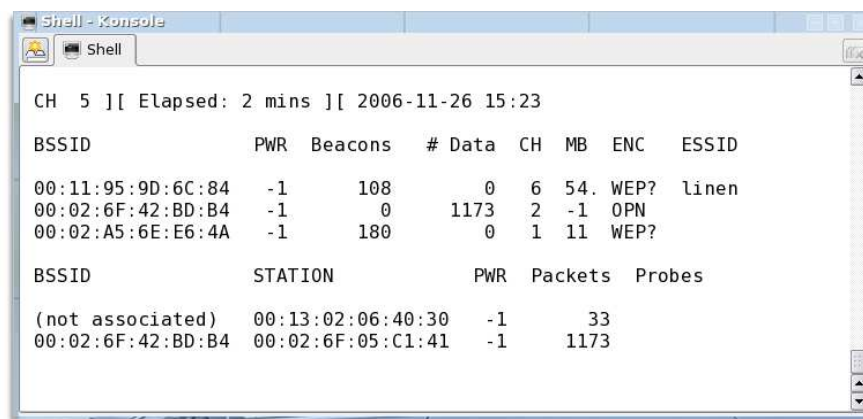
The experiment explained in this section took place at the Nelson Mandela Metropolitan University (NMMU). This is just one of the many experiments conducted while testing the model, and serves as an example. The intrusion tools utilized to perform the various attacks making up the experiment are all tools freely available on the Internet. The tools used are as follows: Airodump, Aireplay, Aircrack, SuperScan and Brutus. Some of the tools utilized are Linux-based, while others are Windows-based.

4.1. The Attacks:

As mentioned previously, the environment in which this experiment occurs is a Microsoft Windows client/server environment, with two windows-based hosts that were known as *Host1* and *Host2* for the experiment. *Host1* was the host installed with the Sentinel IDS wireless sensors and is also the DHCP and FTP server. *Host2* was the host housing the main Sentinel IDS application. *AP1* was the wireless access-point with which *Host1* and *Host2* communicate. *Intruder1* was the malicious intruder running a Linux-based notebook, who wanted to steal information from *Host1*. Now that the background has been given, the next step is to simulate a typical simple attack through an experiment.

Step 1:

First of all, the wireless card of the Linux-based notebook *Intruder1* was placed into monitor mode, allowing it to sniff all wireless traffic within its antenna's range. The network *AP1* was detected while sniffing for wireless signals. Now that *Intruder1* has the access-point's SSID, the next goal is to gain access to the wireless network itself. This sniffing of wireless signal can be seen in Figure 4.



```
CH 5 ][ Elapsed: 2 mins ][ 2006-11-26 15:23
```

BSSID	PWR	Beacons	# Data	CH	MB	ENC	ESSID
00:11:95:9D:6C:84	-1	108	0	6	54	WEP?	linen
00:02:6F:42:BD:B4	-1	0	1173	2	-1	OPN	
00:02:A5:6E:E6:4A	-1	180	0	1	11	WEP?	

BSSID	STATION	PWR	Packets	Probes
(not associated)	00:13:02:06:40:30	-1	33	
00:02:6F:42:BD:B4	00:02:6F:05:C1:41	-1	1173	

Figure 4: Example Sniffing of Wireless Traffic.

Step 2:

To gain access to the wireless network, *Intruder1* had to collect wireless traffic destined for the *AP1*'s SSID. Enough data needed to be collected so that the WEP key could be cracked. To speed up the data collection process, *Intruder1* used Aireplay, which allowed him/her to replay initialization vector packets collected. Once the data had been collected, *Intruder1* runs Aircrack and gains the WEP key, which was "prim8". The cracking of *AP1*'s SSID is depicted in Figure 5.

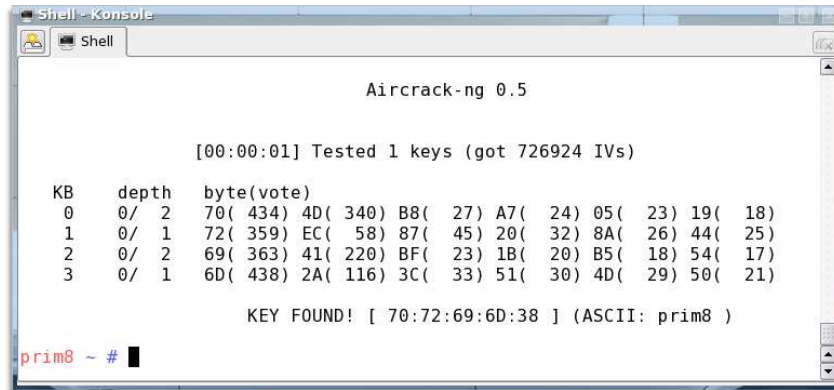


Figure 5: Example Cracking AP1 WEP Key.

Step 3:

After *Intruder1* cracked the WEP key, he/she then proceeded to connect to *AP1* with the newly acquired SSID and WEP Key. This allowed him/her initial access to the organizational network. The first action *Intruder1* took once he/she gained access to the organizational network was to perform a portscan with SuperScan on the network server *Host1*, which was located because it was the DHCP server, and assigned *Intruder1* an IP address when he/she connected. During the portscan of *Host1*, the intruder determined that *Host1* has an FTP server running on it. An example of this can be seen in Figure 6.

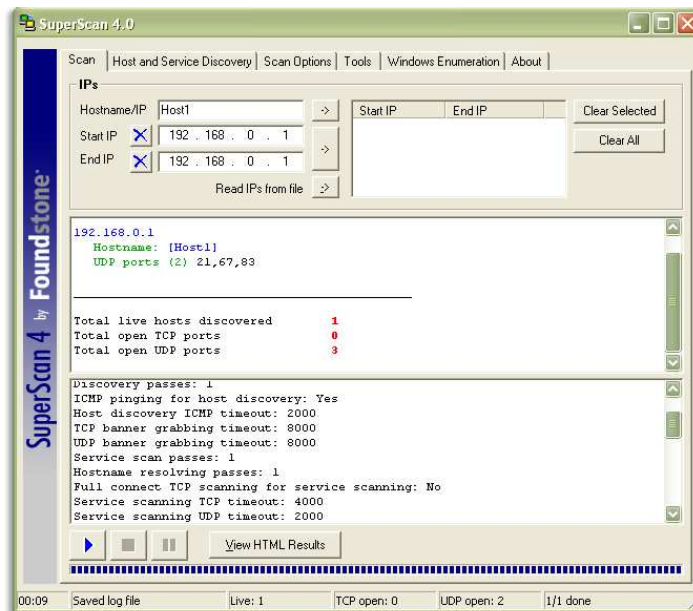


Figure 6: Example Portscan on Host1.

Step 4:

The final part of the attack occurred as *Intruder1* attempted to brute force the administrator accounts password on *Host1* using Brutus. The reason the intruder was unable to complete his/her attack and gain the password, which he/she was attempting to bruteforce, is due to Sentinel IDS sensors on both *Host1* and *Host2* determining that intrusive behaviour had taken place. This can be seen in Figure 7.

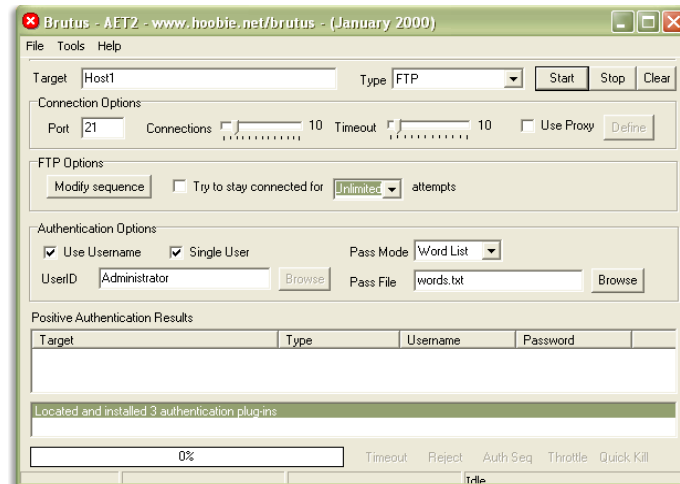


Figure 7: Example Bruteforce Attack on Host1.

The next section will explain the results of the above listed attack as they relate to the three engines. As the output of the two low-level engines is needed for the high-level engine to calculate total attack probability (TPV), the two low-level engine results will be shown and explained first.

4.2. The Results:

After performing the experiment, the next logical step is to analyze and interpret the results. The results of the experiment will be reported according to the individual engine's responses to the various steps in the experiment outlined previously.

Step 1 Result's:

Step 1's Theft of Information attack; saw the intruder placing his/her Linux-based notebook's wireless card into monitor mode (promiscuous mode). Within a minute or so, one of the Sentinel IDS fuzzy engine's wireless generic signatures fired. The signature that fired is one of the probing phase signatures, more specifically a signature which determined that *Intruder1's* MAC address was not listed as a MAC address belonging to any organizational hardware.

This in itself is not seen as to great a threat, as many other organizations in the vicinity may be utilizing wireless networks, but the fact that *Intruder1's* wireless card is in monitor mode suggests that the owner may be sniffing for access-points. The fuzzy engine noted the MAC address of *Intruder1* (00:12:F0:3B:5F:71). This MAC address will be kept for a predetermined time period for correlation purposes. The signature also started a timer to determine the length of time spent in monitor mode by *Intruder1*. The output of the fuzzy engine, at this point, is set at 5% probability of attack. This information was passed by the fuzzy engine to the CAE.

The CAE at this point has only received intrusion probability from the fuzzy engine and after performing the statistical calculations, the CAE output is 2.5% probability of attack. This is too low to implement any active or passive responses.

Step 2 Result's:

Step 2; saw *Intruder1* detecting *AP1* as a viable target and utilizing Airodump to collect as much traffic containing initialization vectors as possible. While *Intruder1* was collecting the data he/she needed to perform the next phase of the attack, two more probing-phase generic wireless signatures fired. The first signature fired as a result of *Intruder1's* wireless card being in monitor mode for a period greater than the limits allowed by the administrator. *Intruder1* spent a total of 3 hours and 5 minutes in monitor mode.

The second wireless generic signature fired during this step of *Intruder1's* attack, fired as a result of an unauthorized MAC address probing *AP1* for its SSID. The fuzzy engine was set to report this as an act of misuse as all organizational wireless hardware have their MAC addresses registered. The output of the fuzzy engine at this point was set at 15% probability of attack due to the two fired generic signatures. The output was once again forwarded to the CAE.

The CAE, at this point, still only received intrusion probability from the fuzzy engine and, after performing the statistical calculations, the CAE output is 7.5% probability of attack. This is still too low to implement any active or passive responses.

Step 3 Result's:

At Step 3 of his/her attack, *Intruder1* had cracked the WEP key for *AP1*, from the data gathered during Step 2. The intruder then proceeded to portscan *Host1*, the DHCP and FTP server, which fired off another two wireless generic signatures.

The first of the signatures fired was an initial-access phase signature, fired during *Intruder1's* first connection to *AP1*, using the SSID and WEP key. This signature had been monitoring *Intruder1's* MAC address for an initial access, due to his/her prior data sniffing and probing activity.

The second signature fired was another probing phase signature, fired due to *Intruder1's* portscan attack. The signature determined that the number and speed at which *Host1's* ports were being accessed were past the limits set by the administrator.

The fuzzy engine's output at this point was set at 40% probability of attack. This information was then reported to the CAE as was done before.

The CAE has still only received intrusion probability from the fuzzy engine at this point. After performing the statistical calculations, the CAE output is 20% probability of attack. This output is enough to cause the CAE to implement a passive response, and the administrator of the network was e-mailed regarding the results.

Step 4 Result's:

The final part of *Intruder1's* attack that he/she was able to commit was Step 4 in which *Intruder1* attempted to bruteforce *Host1's* administrative account for its FTP server.

A hacking-phase system generic signature fired, due to the number of failed login attempts performed by the administrative account. The number of failed login attempts was set in the IDS on setup of the fuzzy engine. At this point, the fuzzy engine's output due to previous signatures fired was set at 75%, and this output once again was forwarded to the CAE.

The neural engine, at this point, plays a role in the detection process. This occurs for two reasons. The first is that the administrative account has had multiple failed login attempts and from the administrator's footprint, the neural engine determines that this is a deviation from his normal behaviour as he mostly enters his/her password in correctly with the first attempt. The second reason the neural engine determined anomalous behaviour for the administrative account is that the invalid password attempts took place during a period of time which is well out of the administrator's usual working hours.

The final piece of anomalous activity detected was due to the fact that the administrator usually works on a network node with a specific MAC address on the wired network, and the failed logon attempts occurred from the wireless network from a host with a MAC address of (00:12:F0:3B:5F:71). This deviated dramatically from the administrator's daily habits. The neural engine's output at this point was set at 70% certainty of attack, and its output was forwarded to the CAE for further action.

At this final step in the attack, the CAE had finally received intrusion probability from the fuzzy engine and the neural engine. After performing the statistical calculations, the CAE output is 72.5% $((70\% + 75\%) / 2)$ probability of attack. This output is enough to cause the CAE to implement both active and passive responses. The passive response was to e-mail the administrator of the network. The active response was to block *Intruder1's* MAC address on all access-points, including *AP1*, and to disable the compromised administrator account on *Host1*.

As can be seen from this experiment, NeGPAIM-W² can identify many different attacks, both on wireless and wired networks. It is thus evident from the experiments performed that NeGPAIM-W² is not just feasible, but has the potential to change the quality of intrusion detection performed on a network as a whole.

5. CONCLUSION

As intrusion attacks increase yearly worldwide, wireless networks also grow in speed, range and capacity making them more inviting to both attackers and organizations. For this reason one needs to implement security measures, allowing for the detection and halting of intrusions before damage can be done.

One such solution is the NeGPAIM-W² IDS, although it may not be the perfect solution, it can go a long way in the protection of an organizations data. The results obtained from the experiments performed on the prototype; show that NeGPAIM-W² is 98% accurate in detection of intrusion attacks, with a false alarm rate of only 2%.

NeGPAIM-W² is an ongoing research project at the NMMU and further research will be performed to increase the effectiveness of the model, to keep up to date with new network and operating system technologies and to improve the model as a whole.

6. REFERENCES

DeYoung, B., Stier, J., & Wampler, L. (2002). *Online Business Alliances: Net Gain @ Speed of Thought!* Retrieved May 25, 2005, from <http://www.cas.nercd.psu.edu/Publications/Webbook/wbChapterDeYoung.pdf>

Gordon, L., Loeb, M., Lucyshyn, W., & Richardson, R. (2005). *2005 CSI/FBI Computer Crime and Security Survey*. Retrieved January 26, 2006, from <http://www.usdoj.gov/criminal/cybercrime/FBI2005.pdf>

Innella, P. (2001). *The Evolution of Intrusion Detection Systems*. Retrieved April 27, 2006, from <http://www.securityfocus.com/infocus/1514>

Intel Corporation. (2005). *Government ICT Policy Primer*. Retrieved February 11, 2007, from <http://www.intel.com/business/bss/industry/government/govtpolicy.pdf>

Lehmann, D. (2005). *What is ID?* Retrieved May 21, 2005, from http://www.sans.org/resources/idfaq/what_is_id.php

Li, Z., Das, A., & Zhou, J. (2005). *Theoretical Basis for Intrusion Detection*. Retrieved February 10, 2007, from <http://www.ntu.edu.sg/home5/pg01316106/Research/PaperIAW.pdf>

LT Consultants & Buck Consultants. (2002). *Final report Comparative survey on urban freight, logistics and land use planning systems in Europe*. Retrieved May 25, 2005, from http://www.ess.co.at/LUTR/PUBLIC/CF_WP1_synthesis.pdf

Telcordia Technologies. (2001). *Number of Internet hosts reaches 100 million*. Retrieved April 16, 2005, from <http://www.telcordia.com/newsroom/pressreleases/01052001.html>

Whitman, E., & Mattford, H. (2003). *Principles of Information Security*. Thomson Course Technology.

Wilhelm, A. (2000). *The state of the Digital Divide in USA*. Retrieved May 25, 2005, from <http://www.digitale-chancen.de/transfer/downloads/MD43.pdf>

Bibliography

- ADC. (2003). *Ethernet Connectivity Selection Tutorial*. Retrieved August 3, 2006, from <http://www.adc.com/Library/Literature/1276553.pdf>
- Aickelin, U., Greensmith, J., & Twycross, J. (2004). *Immune System Approaches to Intrusion Detection – A Review*. Retrieved April 18, 2006, from http://www.cs.nott.ac.uk/~uxa/papers/04icarids_ids_review.pdf
- Airtight Networks. (2006). *Background on Wireless Intrusion Prevention System (WIPS) Patents*. Retrieved August 9, 2006, from <http://www.airtightnetworks.net/news/pdf/background-info-WIPS.PDF>
- Alessandri, D. e. (2001). *Towards a Taxonomy of Intrusion Detection Systems and Attacks*. Retrieved August 8, 2006, from <http://www.maftia.org/deliverables/D3.pdf>
- Alpcan, T., & Bascar, T. (2004). *A Game Theoretic Analysis of Intrusion Detection in Access Control Systems*. Retrieved April 27, 2006, from http://decision.csl.uiuc.edu/~alpcan/papers/alpcan-basar-cdc04_WeA05_6.pdf
- Amato, V. (2000). *Cisco Networking Academy Program: First Year Companion Guide*. Cisco Press.
- Anjum, F., Subhadrabandhu, D., & Sarkar, S. (2004). *Signature based Intrusion Detection for Wireless Ad-Hoc Networks: A Comparative study of various routing protocols*. Retrieved March 21, 2007, from http://www.seas.upenn.edu/~swati/PID21898_final.pdf
- Armosky, B., & Hemenway, M. K. (2000). *Optical Fiber*. Retrieved July 10, 2006, from <http://mcdonaldobservatory.org/teachers/classroom/ttt/OPTIC.pdf>
- Armstrong, P. (2001). *Input-Process-Output Model*. Retrieved March 28, 2007, from <http://www.colostate.edu/Depts/Speech/rccs/theory35.htm>
- Aruba Networks. (2004). *Wireless Intrusion Protection*. Retrieved October 22, 2006, from http://www.unipalm.ie/shadomx/apps/fms/fmsdownload.cfm?file_uuid=41ED5561-E3F6-3B69-EAB0-931D0AD443EC&siteName=unipalm
- Axelsson, S. (2000). *Intrusion Detection Systems: A Survey and Taxonomy*. Retrieved August 10, 2006, from <http://www.mnlab.cs.depaul.edu/seminar/spr2003/IDSSurvey.pdf>
- Bace, R. (1999). *An Introduction to Intrusion Detection Assessment for System and Network Security Management*. Retrieved August 10, 2006, from <http://www.icsalabs.com/icsa/docs/html/communities/ids/whitepaper/Intrusion1.pdf>

- Bace, R. (2000). *Intrusion Detection*. Macmillan Technical Publishing.
- Bace, R., & Mell, P. (2001). *Intrusion Detection Systems*. Retrieved May 16, 2006, from <http://csrc.nist.gov/publications/nistpubs/800-31/sp800-31.pdf>
- Beal, V. (2005). *Intrusion Detection and Prevention, All about IPS and IDS*. Retrieved April 20, 2006, from http://www.webopedia.com/DidYouKnow/Computer_Science/2005/intrusion_detection_prevention.asp
- Becker, F., & Petermann, M. (2005). *Intrusion Detection Systems Elevated to the Next Level*. Retrieved July 6, 2006, from http://events.ccc.de/congress/2005/fahrplan/attachments/560-Paper_IntrusionDetectionSystems.pdf
- Berkan, R. (1997). *Fuzzy System Design Principles (1st ed.)*. U.S.A.: IEEE Press Marketing.
- Bishop, M., & Bailey, D. (1996). *A Critical Analysis of Vulnerability Taxonomies*. Retrieved August 11, 2006, from <http://www.cs.ucdavis.edu/research/tech-reports/1996/CSE-96-11.pdf>
- Blackstock, M., & Sawadsky, N. (2005). *Colligo Security Whitepaper*. Retrieved November 6, 2006, from http://www.colligo.com/_documents/Colligo_Security_Whitepaper.pdf
- Borisov, N., Goldberg, I., & Wagner, D. (2001). *Security of the WEP algorithm*. Retrieved May 30, 2006, from <http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html>
- Botha, M. (2003). *NeGPAIM: Next Generation Proactive Identification Model*. Port Elizabeth, Eastern Cape, South Africa: Port Elizabeth Technikon.
- Boyd, T., & Dasgupta, P. (2004). *Wireless Network Security*. Retrieved July 4, 2006, from <http://cactus.eas.asu.edu/partha/Papers-PDF/2004/WirelessSecurityBoyd.pdf>
- Braue, D. (2003). *Intrusion Detection: Caught in it's own web?* Retrieved September 11, 2005, from http://www.zdnet.com.au/insight/soa/Intrusion_detection_caught_in_its_own_web_/0,39023731,20278214,00.htm
- Broersma, M. (2006). *Test shows how vulnerable unpatched windows is*. Retrieved July 24, 2006, from <http://www.techworld.com/security/news/index.cfm?NewsID=5535>
- Bruneau, G. (2001). *The History and Evolution of Intrusion Detection*. Retrieved April 27, 2006, from <http://www.sans.org/rr/whitepapers/detection/344.php>
- Carter, E. (2002). *Intrusion Detection Systems*. Retrieved September 24, 2006, from <http://www.ciscopress.com/articles/article.asp?p=25334&rl=1>
- Carver, C., & Pooch, U. (2002). *An Intrusion Response Taxonomy and its Role in Automatic Intrusion Resonse*. Retrieved August 14, 2006, from http://www.itoc.usma.edu/marin/Wshop/Papers2000/TP1_4.pdf

- CERT Coordination Center. (2005). *Denial of Service Attacks*. Retrieved May 26, 2005, from http://www.cert.org/tech_tips/denial_of_service.html
- Ciampa, M. (2005). *Security+ Guide to Network Security Fundamentals (Second Edition)*. Massachusetts: Thompson Course Technology.
- Cisco Systems. (2003b). *Deploying layer 2 security in server farms*. Retrieved May 30, 2006, from http://www.cisco.com/application/pdf/en/us/guest/netsol/ns376/c649/cdccont_0900aecd800ebd1e.pdf
- Cisco Systems. (2006). *Security Products at a Glance*. Retrieved July 29, 2006, from http://www.ccard.ru/download/cisco_cpqrq4_en.pdf
- Cisco Systems. (2003a). *Strategies to Protect Against Distributed Denial of Service (DDoS) Attacks*. Retrieved May 26, 2005, from <http://www.cisco.com/warp/public/707/newsflash.html>
- Coffman, K., & Odlyzko, A. (2001). *Growth of the Internet*. Retrieved May 30, 2006, from <http://www.dtc.umn.edu/~odlyzko/doc/oft.internet.growth.pdf>
- CTA. (2002). *Information Security – Intrusion Detection*. Retrieved November 6, 2006, from http://www.cta.com/content/docs/Int_Det.pdf
- defenceindepth.com. (2007). *infosec*. Retrieved March 22, 2007, from <http://www.defenceindepth.com/infosec.html>
- Denning, D. (1986). *An Intrusion Detection Model*. Retrieved April 3, 2007, from <http://www.ece.cmu.edu/~adrian/731-sp04/readings/denning-ids.pdf>
- DeYoung, B., Stier, J., & Wampler, L. (2002). *Online Business Alliances: Net Gain @ Speed of Thought!* Retrieved May 25, 2005, from <http://www.cas.nercrd.psu.edu/Publications/Webbook/wbChapterDeYoung.pdf>
- Doctor, B. (2004). *Intrusion Detection vs. Intrusion Prevention: The Difference and what you need to know*. Retrieved May 26, 2006, from http://www.stillsecure.com/docs/StillSecure_CyberDefense_IPS_v_IDS_0304.pdf
- Dreger, H., Feldmann, A., Paxson, V., & Sommer, R. (2004). *Operational Experiences with High-Volume Network Intrusion Detection*. Retrieved April 26, 2005, from <http://www.icir.org/vern/papers/high-volume-ccs04.pdf>
- DTI UK. (2004). *Achieving Best Practice in your Business*. Retrieved November 23, 2006, from <http://www.dti.gov.uk/files/file9981.pdf>

Du, W., & Mathur, A. (1997). *Categorization of Software Errors that led to Security Breaches*. Retrieved August 11, 2006, from <http://ftp.cerias.purdue.edu/pub/papers/aditya-mathur/du-mathur-categorization.pdf>

Esposito, M., Mazzariello, c., Oliviero, F., Romano, S., & Sansone, C. (2004). *Evaluating Pattern Recognition Techniques in Intrusion*. Retrieved March 20, 2006, from http://www.comics.unina.it/index.php?option=com_docman&task=doc_download&gid=7&Itemid=90&mode=view

Federal Trade Commission. (2004). *Information Compromise and the Risk of Identity Theft: Guidance for Your Business*. Retrieved May 16, 2006, from <http://www.ftc.gov/bcp/online/pubs/buspubs/idthrespond.pdf>

Finn, N. (2001). *Security for 802 access networks: a problem statement*. Retrieved May 11, 2006, from http://grouper.ieee.org/groups/802/3/efm/public/sep02/sec/finn_sec_1_0902.pdf

Flickenger, R. (2002). *Building wireless community networks (First Edition)*. O'Reilly.

Foong Heng, W., Yin Nwe, A., & Ng Hian, J. (2003). *Intrusion Detection in Wireless Ad-Hoc Networks*. Retrieved August 31, 2005, from <http://www.comp.nus.edu.sg/~cs4274/termpapers/0304-l/group4/paper.pdf>

Foster, J. (2006). *Hot Pick: NFR repeats top honours in Intrusion Prevention*. Retrieved September 3, 2006, from http://searchsecurity.techtarget.com/originalContent/0,289142,sid14_gci1170604,00.html

Foundstone Inc. (2007). *SuperScan v4.0*. Retrieved April 2, 2007, from www.foundstone.com/resources/proddesc/superscan.htm

Freer, J. (1988). *Computer Communications and Networks*. Plenum Press.

Gast, M. (2002). *802.11 Wireless Networks: The Definitive Guide*. O'Reilly.

Gast, M. (2004). *Security: Which Layer?* Retrieved May 30, 2006, from http://www.ilabs.interop.net/LANSec/papers/15_Which_layer-LV04.pdf

Goguen, A., Stoneburner, G., & Feringa, F. (2002). *Risk Management Guide for Information Technology Systems*. Retrieved January 15, 2006, from <http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf>

Gonzales, D. V. (2005). *An Analysis of Automated Solutions for the Certification and Accreditation of Navy Medicine Information Assets*. Retrieved May 21, 2006, from http://cistr.nps.navy.mil/downloads/theses/05thesis_gonzales.pdf

Goodman, J. (2003). *How to secure your small to medium sized Microsoft based network: a case study*. Retrieved January 1, 2006, from <http://www.sans.org/rr/whitepapers/basics/1189.php>

Gordon, L., Loeb, M., Lucyshyn, W., & Richardson, R. (2004). *2004 CSI/FBI Computer Crime and Security Survey*. Retrieved April 20, 2005, from http://i.cmpnet.com/gocsi/db_area/pdfs/fbi/FBI2004.pdf

Gordon, L., Loeb, M., Lucyshyn, W., & Richardson, R. (2005). *2005 CSI/FBI Computer Crime and Security Survey*. Retrieved January 26, 2006, from <http://www.usdoj.gov/criminal/cybercrime/FBI2005.pdf>

Guidance Software. (2003). *Incident Response Requirements under ISO 17799*. Retrieved January 13, 2006, from <http://www.guidancesoftware.com/corporate/downloads/whitepapers/ISO17799.pdf>

Hafner, K. (2000). *Difference between Good Hackers and Bad Ones can Often be a Blur*. Retrieved July 11, 2006, from <http://www.thehacktivist.com/archive/news/2000/GoodHackersBadOnes-NYTimes-2000.pdf>

Hansman, S. (2003). *A Taxonomy of Network and Computer Attack Methodologies*. Retrieved March 22, 2007, from antareja.rvs.uni-bielefeld.de/~made/Seminar/Taxonomy/SimonHansmann.pdf

Hash, J. S. (2002). *Risk Management Guidance for Information Technology Systems*. Retrieved May 21, 2006, from <http://csrc.nist.gov/publications/nistbul/02-02.pdf>

Hoobie Inc. (2007). *Brutus FAQ*. Retrieved April 2, 2007, from <http://www.hoobie.net/brutus/>

Hoskins, R. (2006). *PandaLabs Reports 60 Percent of WiFi Networks Are Vulnerable*. Retrieved March 20, 2007, from <http://www.bbwxchange.com/pubs/2006/03/16/page1421-112563.asp>

Howard, J. D. (1997). *An Analysis of Security Incidents on the Internet 1989 - 1995*. Retrieved August 22, 2006, from <http://www.cert.org/research/JHThesis/Start.html>

Hutchinson, K. (2004). *Wireless Intrusion Detection Systems*. Retrieved July 30, 2006, from http://www.sans.org/reading_room/whitepapers/wireless/1543.php

IEEE Computer Society. (2002). *Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) access method and physical layer specifications*. Retrieved May 11, 2006, from <http://standards.ieee.org/getieee802/download/802.3-2002.pdf>

Innella, P. (2002). *Managing Intrusion Detection Systems in Large Organizations, Part Two*. Retrieved March 20, 2007, from <http://www.securityfocus.com/infocus/1567>

Innella, P. (2001). *The Evolution of Intrusion Detection Systems*. Retrieved April 27, 2006, from <http://www.securityfocus.com/infocus/1514>

Intel Corporation. (2005). *Government ICT Policy Primer*. Retrieved February 11, 2007, from <http://www.intel.com/business/bss/industry/government/govtpolicy.pdf>

Interlink Networks. (2002). *A Practical Approach to Identifying and Tracking Unauthorized 802.11 Cards and Access Point*. Retrieved October 22, 2006, from http://www.interlinknetworks.com/graphics/news/wireless_detection_and_tracking.pdf

- IronGeek. (2007). *Hacking Illustrated*. Retrieved April 2, 2007, from <http://www.irongeek.com/i.php?page=security/hackingillustrated>
- ISS. (2002). *Active Wireless Protection*. Retrieved September 2, 2006, from <http://documents.iss.net/whitepapers/ActiveWirelessProtection.pdf>
- ISS. (2006). *RealSecure Network 10/100 Datasheet*. Retrieved July 29, 2006, from http://documents.iss.net/literature/RealSecure/rsn10-100_datasheet.pdf
- ISS. (2001). *Wireless LAN Security 802.11b and Corporate Networks*. Retrieved September 2, 2006, from http://www.iss.net/documents/whitepapers/wireless_LAN_security.pdf
- Jakobsson, M., & Menczer, F. (2005). *Web Forms and Untraceable DDOS Attacks*. Retrieved July 4, 2006, from <http://www.informatics.indiana.edu/fil/Papers/ecb-chapter.pdf>
- Karanth, S., & Tripathi, A. (2004). *Monitoring of Wireless Networks for Intrusions and Attacks*. Retrieved May 10, 2005, from https://www.cs.umn.edu/tech_reports_upload/tr2004/04-010.pdf
- Katzke, S., Ross, R., Johnson, A., Swanson, M., Stoneburner, G., Rogers, G., et al. (2005). *Recommended Security Controls for Federal Information Systems*. Retrieved January 12, 2006, from <http://csrc.nist.gov/publications/nistpubs/800-53/SP800-53>.
- Kawamoto, D. (2005). *Bots slim down to get tough*. Retrieved June 26, 2006, from http://news.zdnet.com/2100-1009_22-5956143.html?tag=nl
- Kemmerer, R. A., & Vigna, G. (2002). *Intrusion Detection: A Brief History and Overview*. Retrieved April 18, 2006, from <http://csdl2.computer.org/comp/mags/co/2002/04/r4s27.pdf>
- Kosko, B. (1993). *Fuzzy Thinking (1st ed.)*. New York, U.S.A: Hyperion Publishing.
- Lampson, B. (2004). Computer Security in the real world. *Computer* 37(6) , 37-46.
- Lehmann, D. (2005). *What is ID?* Retrieved May 21, 2005, from http://www.sans.org/resources/idfaq/what_is_id.php
- Leiner, B., Cerf, V., Clark, D., Kahn, R., Kleinrock, L., Lynch, D., et al. (1997). The Past and Future History of the Internet. *Communications of the ACM* 40(2) , 102-108.
- Lesonsky, R. (2006). *The Entrepreneurs Guide to doing Business Online*. Retrieved July 10, 2006, from <http://www.entrepreneur.com/downloads/paypalbusinessonlineguide.pdf>
- Lesser, J. (2001). *Black Hats Prefer Linux*. Retrieved April 2, 2007, from <http://www.securityfocus.com/columnists/42>
- Li, Z., Das, A., & Zhou, J. (2005). *Theoretical Basis for Intrusion Detection*. Retrieved February 10, 2007, from <http://www.ntu.edu.sg/home5/pg01316106/Research/PaperIAW.pdf>

- Lim, Y., Schmoyer, T., Levine, J., & Owen, H. (2003). *Wireless Intrusion Detection and Response*. Retrieved July 30, 2006, from http://users.ece.gatech.edu/~owen/Research/Conference%20Publications/wireless_IAW2003.pdf
- Lippmann, R., Haines, J., Fried, D., Korba, J., & Das, K. (2000). *The 1999 DARPA Off-Line Intrusion Detection Evaluation*. Retrieved July 28, 2006, from <http://www.ll.mit.edu/IST/ideval/pubs/2000/1999Eval-ComputerNetworks2000.pdf>
- Lockhart, A. (2005). *SNORT-Wireless*. Retrieved September 2, 2006, from <http://snort-wireless.org/>
- Longstaff, T. (2004). *CERT 2004 Annual research report*. Retrieved November 30, 2005, from http://www.cert.org/archive/pdf/cert_rsrch_annual_rpt_2004.pdf
- Lough, D. (2001). *A Taxonomy of Computer Attacks with Application to Wireless Networks*. Retrieved September 1, 2006, from <http://scholar.lib.vt.edu/theses/available/etd-04252001-234145/unrestricted/lough.dissertation.pdf>
- LT Consultants & Buck Consultants. (2002). *Final report Comparative survey on urban freight, logistics and land use planning systems in Europe*. Retrieved May 25, 2005, from http://www.ess.co.at/LUTR/PUBLIC/CF_WP1_synthesis.pdf
- Lynn, T. (2002). *Vulnerability Risk Mitigation – Patching the Microsoft Windows Enviroment*. Retrieved January 1, 2006, from <http://www.sans.org/rr/whitepapers/windows/291.php>
- Maselli, G., Deri, L., & Suin, S. (2002). *Design and Implementation of an Anomaly Detection System: an Empiracal Approach*. Retrieved October 14, 2006, from <http://luca.ntop.org/ADS.pdf>
- McAfee. (2003). *Next Generation Intrusion Detection Systems (IDS)*. Retrieved July 28, 2006, from http://www.mcafee.com/us/local_content/white_papers/wp_intruvertnextgenerationids.pdf
- McCullough, J. (2004). *Caution! Wireless Networking – Preventing a Data Disaster*. Wiley Publishing, Inc.
- McHugh, J., Christie, A., & Allen, J. (2000). *Defending Yourself: The Role of Intrusion Detection Systems*. Retrieved July 6, 2006, from http://www.cert.org/archive/pdf/IEEE_IDS.pdf
- Mell, P., Hu, V., Lippmann, R., Haines, J., & Zissman, M. (2003). *An Overview of Issues in Testing Intrusion Detection Systems*. Retrieved November 26, 2006, from <http://csrc.nist.gov/publications/nistir/nistir-7007.pdf>
- Mirkovic, J., Martin, J., & Reiher, P. (2002). *A Taxonomy of DDOS Attacks and DDoS Defense Mechanisms*. Retrieved July 4, 2006, from http://lasr.cs.ucla.edu/ddos/ucla_tech_report_020018.pdf
- Motorola. (2005). *WiMAX: D vs. E, The Advantages of 802.16e over 802.16d*. Retrieved August 20, 2006, from http://www.motorola.com/networkoperators/pdfs/new/WIMAX_E_vs_D.pdf

Nagaraj, N. (1999). *No Dice*. Retrieved January 8, 2007, from <http://www.hinduonnet.com/businessline/praxis/pr0203/02030040.htm>

NFR. (2006). *NFR Sentivist Overview*. Retrieved July 30, 2006, from <http://www.nfr.com/solutions/download/Sentivist-5-Brochure.pdf>

Nichols, R., & Lekkas, P. (2002). *Wireless Security: Models, Threats and Solutions*. McGraw Hill.

Ning, P., Du, W., & Fang, L. (2005). *LAD: Localization Anomaly Detection for Wireless Sensor Networks*. Retrieved November 16, 2006, from www.cis.syr.edu/~wedu/Research/paper/lad_ipdps05.pdf

Owens, L., & Karygiannis, T. (2002). *Wireless Network Security 802.11, Bluetooth and Handheld Devices*. Retrieved May 30, 2006, from http://csrc.nist.gov/publications/nistpubs/800-48/NIST_SP_800-48.pdf

Panduit. (2004). *The Evolution of Copper Cabling Systems from Cat5 to Cat5e to Cat6*. Retrieved August 3, 2006, from <http://www.panduit.com/products/WhitePapers/098765.pdf>

Patzakis, J. (2003, September). *New Incident Response Best Practices*. Retrieved November 30, 2006, from <http://www.guidancesoftware.com/downloads/IRBestPractices.pdf>

Paulauskas, N., & Garsva, E. (2006). *Computer System Attack Classification*. Retrieved August 14, 2006, from <http://www.ktu.lt/lt/mokslas/zurnalai/elektr/z66/1392-1215-2006-02-66-84.pdf>

Petrovic, S. (2005). *Vulnerabilities in wireless networks and intrusion detection*. Retrieved April 18, 2006, from http://www.telenor.com/elektronikk/volumes/pdf/1.2005/Page_086-091.pdf

phenomenologycenter.com. (1997). *What is Phenomenology?* Retrieved March 21, 2007, from <http://www.phenomenologycenter.org/phenom.htm#4>

Radvanovsky, B. (2004). *Whitepaper: Hiding an Intrusion Detection System (IDS)*. Retrieved May 21, 2006, from <http://www.unixworks.net/papers/wp-003.pdf>

Ranum, R. (2003). *False Positives: A User's Guide to Making Sense of IDS Alarms*. Retrieved June 6, 2005, from <http://www.icsalabs.com/html/communities/ids/whitepaper/FalsePositives.pdf>

Rogers, L. R. (2001). *Everyone's a System Administrator*. Retrieved May 21, 2006, from http://www.sei.cmu.edu/news-at-sei/columns/security_matters/2001/3q01/security-3q01.pdf

Rogers, L. R. (2004). *The Goal of Computer Security or What's Yours is Yours Until You Say Otherwise!* Retrieved May 21, 2006, from http://www.sei.cmu.edu/news-at-sei/columns/security_matters/2004/2/security-matters-2004-2.pdf

SafeNet. (2006). *Securing Fiber Optic Communications*. Retrieved July 10, 2006, from <http://www.safenet-inc.com/library/8/SecuringFiberOpticComm.pdf>

Salmanian, M., Lefebvre, J. H., Leonard, S., & Knight, S. (2004). *Intrusion Detection in 802.11 Wireless Local Area Networks*. Retrieved October 26, 2006, from <http://www.ottawa.drdc-rddc.gc.ca/docs/e/TM2004-120.pdf>

Schiesel, S. (2005). *Growth of wireless internet opens new path for thieves*. Retrieved November 21, 2006, from <http://www.nytimes.com/2005/03/19/technology/19wifi.html?ex=1268888400&en=51d90e7518bba5d6&ei=5090&partner=rssuserland>

Shadow Team. (2003). *SHADOW Version 1.8 Installation Manual*. Retrieved July 30, 2006, from <http://www.nswc.navy.mil/ISSEC/CID/SHADOW-1.8-Install.pdf>

Sheldon, T. (2001). *Cable and Wiring*. Retrieved July 24, 2006, from <http://www.linktionary.com/c/cabling.html>

SRI. (2006). *Event Monitoring Enabling Responses to Anomalous Live Disturbances (EMERALD)*. Retrieved July 30, 2006, from <http://www.sdl.sri.com/projects/emerald/>

Stallings, W. (1997). *Data and Computer Communications*. Prentice-Hall, Inc.

Stubblefield, A., Ioannidis, J., & Rubin, A. (2002). *Using the Fluhrer, Mantin, and Shamir Attack to Break WEP*. Retrieved August 20, 2005, from <http://www.isoc.org/isoc/conferences/ndss/02/proceedings/papers/stubbl.pdf>

Stylios, C., Georgopoulos, V., & Groumpos, P. (1997). *The Use of Fuzzy Cognitive Maps in Modeling Systems*. Retrieved October 13, 2006, from <http://med.ee.nd.edu/MED5/PAPERS/067/067.PDF>

Symantec. (2006a). *Symantec Netproowler 3.5*. Retrieved July 29, 2006, from <http://www.symantec.com/region/can/eng/product/np/>

Symantec. (2006b). *W32.Sober Worm Removal Tool*. Retrieved January 26, 2006, from <http://securityresponse.symantec.com/avcenter/venc/data/w32.sober.removal.tool.html>

Tan, K., Killourly, K., & Maxion, R. (2002). *Undermining an Anomaly-Based Intrusion Detection System Using Common Exploits*. Retrieved October 26, 2006, from <http://www.cs.cmu.edu/afs/cs.cmu.edu/user/maxion/www/pubs/TanKillourhyMaxion02.pdf>

Telcordia Technologies. (2001). *Number of Internet hosts reaches 100 million*. Retrieved April 16, 2005, from <http://www.telcordia.com/newsroom/pressreleases/01052001.html>

Tripwire. (2006). *Tripwire Enterprise Datasheet*. Retrieved July 29, 2006, from http://www.tripwire.com/files/literature/product_info/Tripwire_Enterprise_Overview.pdf

Unisys. (2005). *Access Control Solutions*. Retrieved July 24, 2006, from http://www.unisys.com/eprise/main/admin/micro/doc/Access_Control_Solutions.pdf

- Valeur, F., Vigna, G., Kreugel, C., & Kemmerer, R. (2004). *A Comprehensive Approach to Intrusion Detection Alert Correlation*. Retrieved October 21, 2006, from http://thor.auto.tuwien.ac.at/~chris/research/doc/tdsc04_correlation.pdf
- Vaughan-Nichols, S. (2004). Achieving Wireless Broadband with WiMAX. *Computer* 37(6) , 10-13.
- Vijayaraghavan, G. (2003). *A Taxonomy of E-Commerce Risks and Failures*. Retrieved August 15, 2006, from http://www.testingeducation.org/articles/ecommerce_taxonomy.pdf
- Wang, C., & Knight, J. (2000). *Towards Survivable Intrusion Detection*. Retrieved July 28, 2006, from <http://www.cert.org/research/isw/isw2000/papers/38.pdf>
- Wardriving.com. (2002). *Wardriving HOWTO (Un-official)*. Retrieved March 20, 2007, from <http://www.wardriving.com/doc/Wardriving-HOWTO.txt>
- Weimann, G. (2004). *Cyber Terrorism, How Real is the Threat?* Retrieved January 26, 2006, from <http://www.usip.org/pubsspecialreportssr119.pdf>
- West-Brown, M., Stikvoort, D., Kossakowski, K., Killcrece, G., Ruefle, R., & Zajicek, M. (2003). *Handbook for Computer Incident Response Teams (CSIRTs)*. Retrieved January 13, 2006, from www.cert.org/archive/pdf/csirt-handbook.pdf
- White, G., & DiCenco, D. (2005). *Information sharing needs for national security*. Retrieved November 26, 2005, from <http://csdl2.computer.org/comp/proceedings/hicss/2005/2268/05/22680125c.pdf>
- Whitman, E., & Mattford, H. (2003). *Principles of Information Security*. Thomson Course Technology.
- Wi-Fi Alliance. (2005). *Deploying Wi-Fi Protected Access (WPA) and WPA2 in the Enterprise*. Retrieved November 6, 2006, from http://www.wi-fi.org/files/uploaded_files/wp_9_WPA-WPA2%20Implementation_2-27-05.pdf
- Wilder, F. (1998). *A Guide to the TCP/IP Protocol Suite, Second Edition*. Artech House, INC.
- Wilhelm, A. (2000). *The state of the Digital Divide in USA*. Retrieved May 25, 2005, from <http://www.digitale-chancen.de/transfer/downloads/MD43.pdf>