Intrusion Detection Systems –
Technologies, Weaknesses and Trends

Examensarbete utfört i Informationsteori

av

Martin Arvidson

Markus Carlbark

LITH-ISY-EX-3390-2003

Stockholm 2003

Intrusion Detection Systems –
Technologies, Weaknesses and Trends

Examensarbete utfört i Informationsteori

av

Martin Arvidson

Markus Carlbark

LITH-ISY-EX-3390-2003

Handledare: Michaela Fatouros

Examinator: Viiveke Fåk

Stockholm 2003-02-25

Avdelning, Institution
Division, Department

Institutionen för Systemteknik
581 83 LINKÖPING

Datum
Date
2003-02-25

Språk
Language

Svenska/Swedish
X Engelska/English

Rapporttyp
Report category

Licentiatavhandling
X Examensarbete

C-uppsats
D-uppsats

Övrig rapport___

ISBN

ISRN LITH-ISY-EX-3390-2003

Serietitel och serienummer **ISSN**
Title of series, numbering

URL för elektronisk version
http://www.ep.liu.se/exjobb/isy/2003/3390/

Titel
Title
Intrångsdetekteringssystem - Teknologier, Svagheter och Trender
Intrusion Detection Systems - Technologies, Weaknesses and Trends

Författare
Author
Martin Arvidson, Markus Carlbark

Sammanfattning
Abstract
Traditionally, firewalls and access control have been the most important components used in order to secure servers, hosts and computer networks. Today, intrusion detection systems (IDSs) are gaining attention and the usage of these systems is increasing. This thesis covers commercial IDSs and the future direction of these systems. A model and taxonomy for IDSs and the technologies behind intrusion detection is presented.
Today, many problems exist that cripple the usage of intrusion detection systems. The decreasing confidence in the alerts generated by IDSs is directly related to serious problems like false positives. By studying IDS technologies and analyzing interviews conducted with security departments at Swedish banks, this thesis identifies the major problems within IDSs today. The identified problems, together with recent IDS research reports published at the RAID 2002 symposium, are used to recommend the future direction of commercial intrusion detection systems.

Nyckelord
Keyword
computer security, IDS, intrusion detection, taxonomy, weaknesses

Abstract

Traditionally, firewalls and access control have been the most important components used in order to secure servers, hosts and computer networks. Today, intrusion detection systems (IDSs) are gaining attention and the usage of these systems is increasing. This thesis covers commercial IDSs and the future direction of these systems. A model and taxonomy for IDSs and the technologies behind intrusion detection is presented.

Today, many problems exist that cripple the usage of intrusion detection systems. The decreasing confidence in the alerts generated by IDSs is directly related to serious problems like false positives. By studying IDS technologies and analyzing interviews conducted with security departments at Swedish banks, this thesis identifies the major problems within IDSs today. The identified problems, together with recent IDS research reports published at the RAID 2002 symposium, are used to recommend the future direction of commercial intrusion detection systems.

Abstract

Table of Contents

List of Figures

1 Introduction

This thesis covers the different technologies used in today's commercial intrusion detection systems (IDSs), identifies the main problems with these technologies and covers recent research advances within the IDS area. Seven major vendors of IDS products and some recent research reports published in [3] are presented and reviewed. Finally, the result of interviews carried out with the major banks in Sweden will be presented. This result will include how the IT-security department at these banks looks at today's IDSs. All of this information is used to recommend the future direction of commercial IDSs.

1.1 Background

The idea of a system that detects intrusions in computer networks has been around for more than two decades. One of the earliest papers about intrusion detection is James P. Anderson's *Computer Security Threat Monitoring and Surveillance*, published in 1980. At first, intrusion detection mainly was a research subject and not something found in the commercial market. However, in the last 10 years the commercial market has begun to grow and today's products are getting more and more advanced. Although there have been a lot of improvements in the IDS field during the last years there are some major problems left for the vendors and researchers to solve. Therefore, this thesis presents the result of an analysis regarding IDS products and their main problems today.

1.2 Methodology

This section describes how the research for this thesis was carried out, the sources of information used and how the interviews were conducted.

1.2.1 Research

The background work for this thesis covered a lot of basic material about intrusion detection. It was conducted in order to find the origin of IDS as well as what IDS is today. During this phase technical reports, articles and some books covering the area were read. To get some basic understanding of the commercial market, a trade show called *"Computer networks & info security 2002"* was visited. At this time, an identification of the goals of this thesis was made.

This thesis describes the technologies that exist in today's products, it does not describe products individually. Keep in mind that this is by no means any kind of recommendation for a specific product.

As stated before, the research for chapter 2-4 was initiated by reading general background material about IDS. After this, a suitable model was created and a decision to use an existing taxonomy found in [4] was made. This taxonomy was later adjusted to better fit the purpose of this thesis. The products covered in chapter 4 were selected based on the magic quadrant in [5]. The material read for chapter 4 included vendors' material posted on their websites and some independent product surveys. This material combined with previous background knowledge was used to describe the different technologies found in the products. The information in this thesis has not been verified

with any practical experience. None of the described programs has been tested or tried out in any way.

Chapter 5 covers the main problems identified within the different areas of the taxonomy. These problems have been identified mainly by using knowledge acquired in the early stages of the research work.

Chapter 6 presents the result of the interviews conducted with Swedish banks. A description of the interviews and how this material is presented is covered in section 1.2.2.

Chapter 7 covers recent advances in intrusion detection research. These advances have been presented at the RAID symposium 2002 held in Zurich, and the reviewed papers can be found in [3]. This material was covered since RAID is a respected symposium and these papers have been selected by well-known researchers within the IDS community.

1.2.2 Interviews

This thesis presents the view on intrusion detection of some larger Swedish corporations. Since it was preferable that these corporations had knowledge and experience from intrusion detection, a decision was made to interview the Swedish banks. This decision was based on the fact that the banks are very conscious of security and therefore probably are aware of, and have experience from, IDS products.

The identification of the single banks to interview was not very hard since there are few large banks in Sweden. Four major and three smaller banks were asked for an interview. Unfortunately, not all banks could participate, but the banks that took part in the interviews are:

- Handelsbanken
- Föreningssparbanken
- SEB
- Östgöta Enskilda Bank (Danske Bank)
- Ikanobanken

Personal interviews with members of the IT-security department at each bank were conducted. Before the interviews, the questions found in appendix A were sent out. The interviews were carried out in an open way and the questions were merely used to start and keep the focus of the discussion. The first part of the questions aimed at gaining a deeper understanding of the corporation's IT-security department. Information about the type of IDS used (if any) and what kind of decisions they had taken regarding this issue was also acquired. The last part of the interview aimed at understanding their challenges and the main problems of IDS according to them. They also stated what they would like to see from the IDS market in the future.

This report does not state which bank said what. It merely summarizes the results of the interviews. It cannot show the answers that each specific bank has given, due to confidentiality reasons. Also, the banks may have declined to answer some questions and if so, this will not be shown in the summary.

1.3 Document Outline

The report has been divided into nine chapters. The content in each chapter is:

1. **Introduction** contains an introduction to the subject and a description on how the work was carried out.
2. **Computer Security Background** covers basic computer security.
3. **Intrusion Detection Background** presents our model of an IDS and a fitting taxonomy. The model and the taxonomy are used in later chapters to classify the different parts of an IDS. A basic intrusion detection terminology is also presented in this chapter.
4. **IDS Today** covers IDS solutions from seven major vendors. A description of the different technologies in these products is made and the taxonomy presented in chapter 3 is used to classify these technologies.
5. **Problems and Challenges** presents the main problems and challenges with the technologies covered in chapter 4.
6. **Result of the Interviews** covers the result of the interviews.
7. **Recent Research Advances** presents recent advances within research found in [3].
8. **Conclusions and recommendations** summarizes this thesis and presents conclusions, recommendations and future work.
9. **Bibliography** contains the material we have referenced in this thesis.

A. **Appendix A** contains the questions used during the interviews.

B. **Appendix B** covers abbreviations used in the thesis.

C. **Appendix C** lists the technologies used by the different vendors.

1.4 Target Audience

This thesis provides good reading for those who want to find out more about intrusion detection and what the market of intrusion detection offers today. Anyone who is planning to invest in IDS equipment can use this thesis to improve their understanding of IDS products. In addition, a summary of recent research advances is given and this material can provide a good starting point for anyone interested in IDS research. Finally, developers of IDSs and researchers can benefit from reading chapters 5 to 8.

The reader should have some basic knowledge of computer networks and security. An introduction to these areas is presented in chapter 2 but this introduction does not cover everything needed to fully understand this thesis.

2 Computer Security Background

This chapter covers the parts of computer security most necessary for understanding the following chapters of this thesis.

2.1 A Secure Computer System

"A secure computer system is a system that can be depended upon to behave as it is expected to do" [6].

In order to achieve this, the components that make up the system must, at some point, be trusted. First, the hardware has to be trusted to behave as expected, thus minimizing the possibility of hardware failure. Second, the software installed and running on the system must be trusted to behave as expected and third, the users of the system must be trusted to behave as expected. Even if all of the above is true, everyone who could gain access to the system (in the case when the computer is connected to the Internet, probably the whole world) have to be trusted to behave as expected.

Since trust is a delicate virtue and often abused, a way to protect our computer systems, detect any malicious activity and react upon the detection must be found. This leads us to the basic definition of protective measures [7]:

- **Prevention**
  Take measures that prevent your assets from being damaged.
- **Detection**
  Take measures that allow you to detect when an asset has been damaged, how it has been damaged, and who caused the damage.
- **Reaction**
  Take measures that allow you to recover your assets or to recover from a damage to your assets.

If the time an asset (e.g. computer system) can be protected is greater than the time it takes to detect and react upon an incident, the security for the asset is sufficient.

2.2 Traditional Computer Security

Computer security can be divided into three cornerstones [7]:

- **Confidentiality**
  Prevention of unauthorized disclosure of information.
- **Integrity**
  Prevention of unauthorized modification of information.
- **Availability**
  Prevention of unauthorized withholding of information or resources.

The following definition specifies how intrusion detection fits in this categorization of security.

"Intrusion detection is the process of identifying and responding to malicious activities targeted at computing and networking resources." [8]

Here the *malicious activities* refer to actions that jeopardize the confidentiality, integrity or availability of information or resources. An intrusion detection system (IDS) is hence "a computer system (possibly a combination of software and hardware) that attempts to perform intrusion detection". [9]

Using the definition for protective measurements in section 2.1, an IDS is concerned with the detection and reaction part of the definition.

2.3 Threats

A threat to a computer system is defined as "any potential occurrence, malicious or otherwise, that can have an undesirable effect on the assets and resources associated with a computer system". [14]

The entities that perform malicious activities, defined as *intruders*, can be categorized according to James P. Anderson's threat matrix [10]:

	Penetrator not authorized to use data/program resource	Penetrator authorized to use data/program resource
Penetrator not authorized use of computer	External penetration	-
Penetrator authorized use of computer	Internal penetration	Misfeasance

Figure 1: Anderson's threat matrix.

Although this classification was created in 1980 for the U.S. Air Force and deals with intrusion detection in a super computer environment, it is still very useful when describing the possible threats toward computer resources. To bring the classification up to date some of the terms have to be redefined. *Use of computer* stated above both includes physically accessing a computer (e.g. login on a terminal) as well as remotely accessing a computer (e.g. login via the Internet). Also the *data/program resource* category includes any type of information stored or transmitted electronically as well as services running on a computer or network device.

2.3.1 External Penetration

In this scenario the intruder is not authorized to use neither the computer nor any data/program resources. As an example the intruder might be an employee of a corporation who wants to access the private intranet of a rival company to steal customer specific information from their customer database. In order to accomplish this he must probably break through the company firewall and gain root access to one or more internal servers. This intrusion is the easiest to detect with today's products.

2.3.2 Internal Penetration

In this scenario the intruder is authorized to use the computer but is not authorized to use the data/program resource. As an example the intruder could be an employee with access to the company intranet through remote access or by his office terminal. The intruder wants to access the company's customer files (to which he has no access) in order to sell these to a rival company. Since he has access to the intranet it is easier for him to accomplish this, and also much harder to detect, than in the example above.

2.3.3 Misfeasance

In this scenario the intruder is both authorized to use the computer and the data/program resource. As an example the intruder could be an executive employee with full company access who wishes to sell customer information to a rival company. This intrusion, when a user abuses his privileges, is most difficult to detect.

2.4 Vulnerabilities

A vulnerability of a computer system is "some unfortunate characteristic that makes it possible for a threat to potentially occur". [14]

Vulnerabilities are weaknesses in computer systems that an intruder could exploit for personal gain. Since no computer system can be totally secure the second best is to successfully identify the existing vulnerabilities. Vulnerabilities can be divided in the following categories. [6]

2.4.1 Design

Many vulnerabilities in computer systems are due to security holes in the hardware or software components. With the increasing competition among developers of hardware equipment, operating systems and applications, there is not always time to thoroughly test the products before releasing new versions to the market. This may leave several vulnerabilities in the form of weak services or back doors that can be exploited and used in order to gain access to a system. In time, the vendors will provide patches or upgrades for their products, but until then the systems are open to intrusions.

2.4.2 Management

Even a fairly secure computer system can have vulnerabilities if the operator lacks sufficient knowledge when configuring it. Setting up the firewall in an insecure way could let everyone through to the company intranet. And making an error when giving user rights in an operating system could let anyone logon to the computer with root access.

2.4.3 Trust

As was discussed in the beginning of this chapter, trust is a delicate virtue that, when given lightly, can be compared with ignorance. Trusting a computer component to behave as expected is one issue, trusting someone with your intranet password is another. If an intruder wants to get access to a resource on your network drive, he could call you pretending to be the company helpdesk and claim that he needs your

password. By using social engineering skills and gaining people's trust, an intruder can save himself a lot of time.

2.5 Attacks

An *attack* on a computer system is "some action taken by a malicious intruder that involves the exploitation of certain vulnerabilities in order to cause an existing threat to occur". [14]

The number of different attacks is vast and no further definitions will be given since it falls outside the scope of this thesis.

Whichever vulnerability a potential intruder tries to exploit, it is up to the intrusion detection system to successfully detect any attacks, thus keeping your computer systems and network environment as secure as possible.

3 Intrusion Detection Background

This chapter covers basic terminology, an IDS model and a taxonomy for IDSs. The model describes the important parts of an IDS and the taxonomy is used to classify different technologies later in this thesis.

3.1 Basic IDS Terminology

- **Console**: The user interface of an IDS.
- **Denial of Service**: A type of attack. When a host that offers a service, e.g. a web server, is attacked in a way that prevents it from continuously offer the service, the host is victim to a denial of service attack.
- **Event**: The internal message of an IDS.
- **False negative**: When no alarm is generated although there is an attack.
- **False positive**: When an alarm is generated although there is no attack. Also known as false alarm.
- **Kernel**: The kernel is the essential center of a computer operating system, the core that provides services for all other parts of the operating system.
- **Manager**: Collects events generated from one or more sensors and performs an analyzing scheme.
- **HIDS** (Host-based Intrusion Detection System): An IDS that monitors one host, usually the one it is installed on.
- **IPS** (Intrusion Prevention System): An IDS that uses proactive responses as response method.
- **Hybrid IDS**: A HIDS that uses, in addition to other information sources, network packets as audit source.
- **NIDS** (Network-based Intrusion Detection System): An IDS that monitors network traffic on the network segment at which it resides.
- **Promiscuous mode**: A NIC (network interface card) in this mode captures all passing network traffic.
- **Sensor**: This is the smallest and most basic variant of an IDS. Although it can function on its own, sensors are usually used as the information gathering part in larger system configurations.
- **Signature**: The formula that describes an attack. The exact notion may vary depending on the IDS, but generally it follows the IF-THEN-ELSE structure.

Attack signatures in IDSs are very similar to virus signatures in anti-virus applications.

- **Snort**: An open-source network-based IDS.

3.2 An IDS-model

Intrusion detection systems exist in a multitude of configurations and there are many different opinions and designations on what an IDS looks like. An existing model, that was both complete and generic, could not be found. A generic model described in [6] was first used but it lacked modularity, a clear flow of information and logically named components. The extended model shown in figure 2 was developed to deal with these shortcomings.

Each box is considered as a stand-alone component, which performs a single task. The arrows describe the flow of information. Some of the boxes have a lighter outline. These boxes are not necessary for the IDS to be operational, although almost every IDS today utilizes them. All components can reside in the same physical system but it is also possible for them to be deployed separately. The components are described in detail in the following sections.

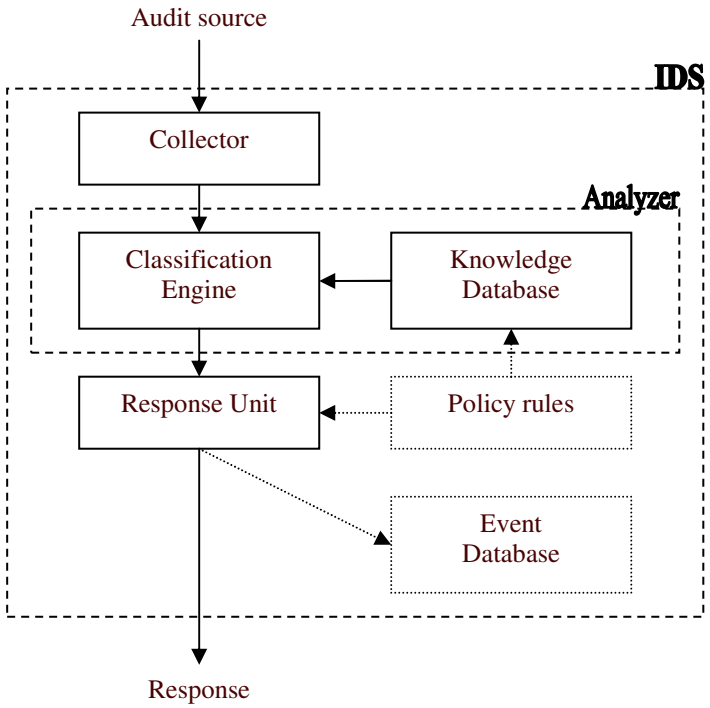

Figure 2: An IDS model.

3.2.1 Audit Source

The audit source is the input to the IDS, the raw data, which can have several different formats depending on the type of IDS and where the IDS is located. Examples of audit sources are application logs, IP-packets and the output from other IDSs. A more thorough description of audit sources will be given in section 3.3.1.

3.2.2 Collector

The collector samples the audit source, either in real-time or periodically, and pre-processes the information. The pre-processing includes transformation of the sampled information into an internal standard format, known by the analyzer. A preliminary reduction of data, e.g. the grouping of similar log entries, is often a part of the pre-processing step. If the IDS is monitoring some kind of connection-oriented protocol, the collector may cache the network packets for session reconstruction.

3.2.3 Analyzer

The analyzer consists of a classification engine and a knowledge database. The function of the analyzer is to determine if the data sent by the collector contains signs of an attack. When an attack is found, the analyzer produces one or more events that are passed on to the response unit.

3.2.3.1 Knowledge Database

The knowledge database is the long term memory of the IDS. It contains detailed attack information that varies depending on the type of IDS.

3.2.3.2 Classification Engine

The classification engine tries to determine if the data received from the collector is proof of an attack. In general it does this by comparing the data with the information stored in the knowledge database according to one or more detection methods. The different methods will be discussed in section 3.3.2. If signs of an attack are found, an event is constructed containing all the relevant attack-related information. The event is usually classified according to the severity of the attack and then passed on to the response unit.

3.2.4 Response Unit

The response unit decides which actions to perform depending on the incoming events and the level of severity. A wide variety of different responses exists and these are discussed in detail in section 3.3.3.

3.2.5 Policy Rules

Policy rules allow us to configure how the IDS should perform detection and react to intrusions. It does this by letting us select a subset of the knowledge database to use in the analyzer and choosing which responses a certain event should trigger in the response unit. Since this feature is optional, an IDS without this module would always use the whole knowledge database for intrusion detection and always respond to attacks in a predefined way.

3.2.6 Event Database

The event database is where all the event information produced by the IDS is stored. The decision if an event should be logged is controlled by the policy and it is taken in the response unit. The database can later be used in a multitude of ways (e.g. doing exhaustive searches or for generating reports of attack statistics).

3.3 Taxonomy

Some kind of framework is needed in order to categorize different IDS solutions. This framework should make it easy to identify and categorize the different technologies of an IDS. The first taxonomy presented here is an already existing taxonomy taken from [4]. This taxonomy has been extended to better fit the needs of this thesis. The original taxonomy is presented in figure 3.

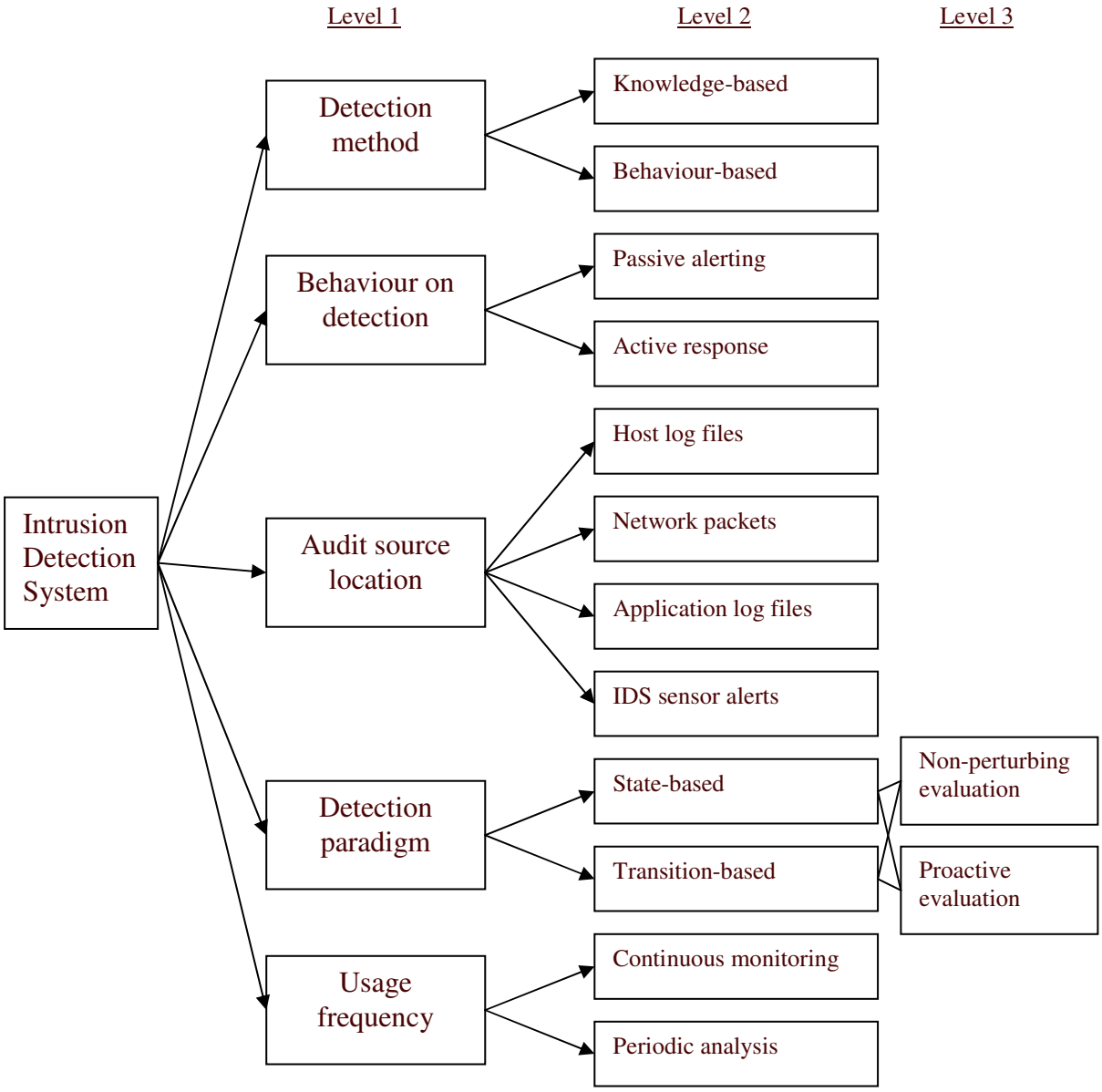

Figure 3: Debar's IDS taxonomy.

The extended taxonomy, which is used in the rest of this thesis, is shown in figure 4.

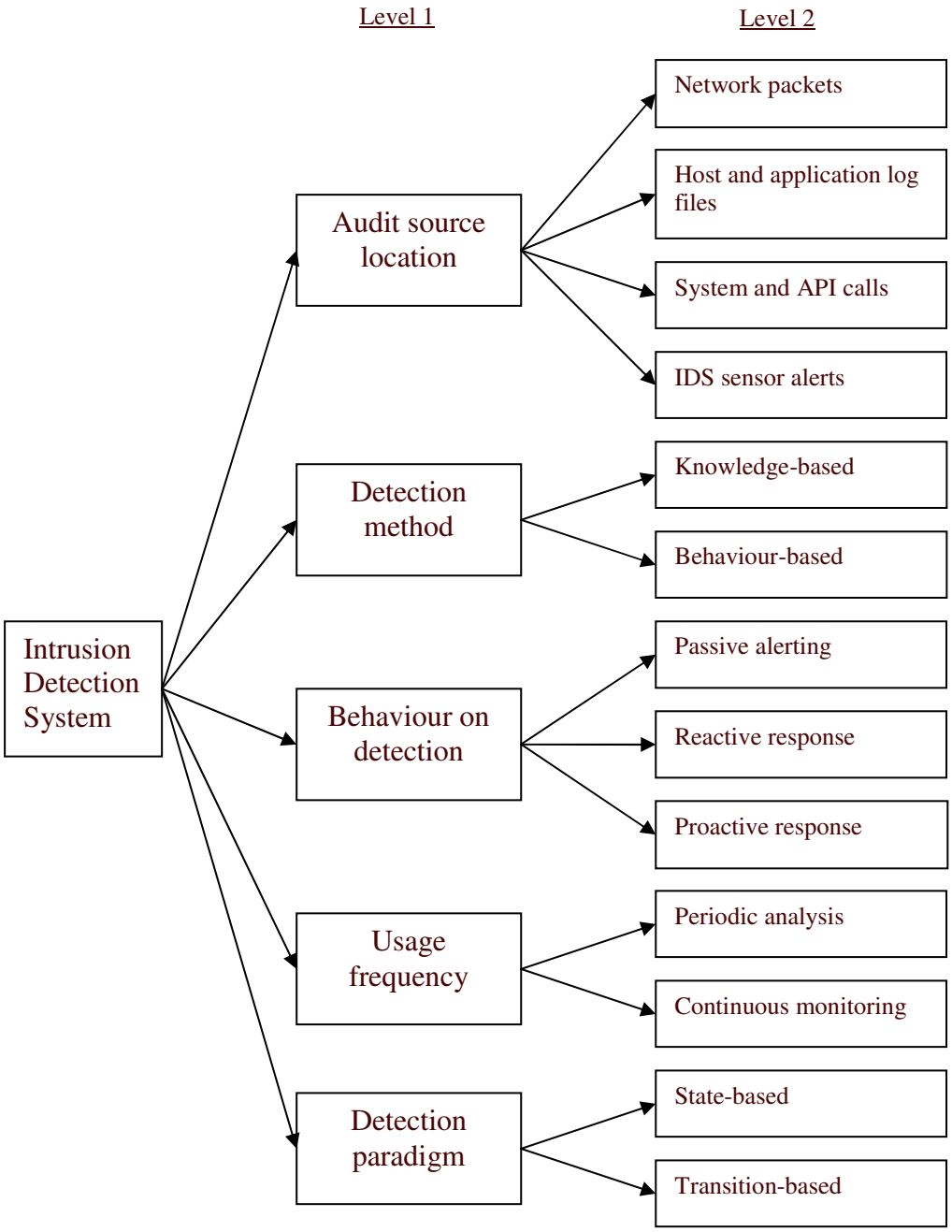

Figure 4: Modified version of the taxonomy.

In the extended taxonomy, some level 2 categories have been added and the order of the level 1 categories have been changed. The order was changed so that it better follows the order of the model (the model starts with the input to the IDS, the audit source). One type of audit source, system and API calls, was added since products that use this exist. In addition, the host log files and the application log files categories have been combined since they are very similar. One type of behaviour on detection, active response, has been divided into two types; reactive and proactive response. This was done since these two categories earlier fell in the same class even though they are quite different.

It should be possible to classify an IDS in all of the level 1 categories. For example, an IDS has to have a way to detect attacks, a detection method, and so far there are only two different kinds of detection methods; behaviour-based and knowledge-based. As long as no new technology appears, there should be no problem to decide whether a detection method is behaviour-based, knowledge-based or both. Note that an IDS has to have at least one of the level 2 categories for every level 1 category. Below, the difference between the different categories is explained.

3.3.1 Audit Source Location

There are different ways for an IDS to collect its data. The input to the IDS could be either log files, packets from the network, system and API calls or events from another IDS.

3.3.1.1 Network Packets

If the IDS is network-based (NIDS), the packets are collected from the network. This is usually done by putting one network card of the machine that the NIDS reside on in promiscuous mode and thereby letting the NIDS see all the passing traffic.

3.3.1.2 Host and Application Log Files

When a host-based IDS (HIDS) is in use, the input is most often received from an OS or application log file. These systems are usually applied to important servers, but can also be placed at firewalls to watch the firewall log and alert security officers when something out of the ordinary happens at the firewall.

3.3.1.3 System and API Calls

Another type of input that a HIDS could have is system and API calls. A HIDS could reside between the kernel and any other application, looking at all the system calls trying to find (and possibly stop) suspect system calls. In this way, the HIDS could detect malicious behaviour of programs as well as users.

3.3.1.4 IDS Sensor Alerts

The last kind of input is events from another IDS. This is common in larger systems where you usually have some NIDS and some HIDS reporting to a central IDS, which analyzes all the events from the different IDSs.

3.3.2 Detection Method

The detection method describes how the system detects events. There are two ways of doing this; knowledge-based or behaviour-based.

3.3.2.1 Knowledge-based

With this method, the system has some kind of knowledge about how attacks look. This means that everything the system does not explicitly recognize as an attack is considered normal. This is usually solved by using signatures to recognize attacks. This method can be very precise (depending on the signature) and therefore should have a relatively low false positive rate.

3.3.2.2 Behaviour-based

If the detection method is behaviour-based the IDS is trying to detect bad behaviour by knowing what normal behaviour looks like. If anything that is not considered normal is detected, the IDS signals that it has detected an attack.

3.3.3 Behaviour on Detection

Behaviour on detection, or responses, are actions taken by the IDS as a result of a generated event. The taxonomy in [4] divides responses into active or passive. Due to the introduction of intrusion prevention systems (IPSs) the active category is further divided into proactive and reactive. IPSs are intrusion detection systems that try to prevent an attack by using proactive responses.

3.3.3.1 Passive Alerting

Passive alerting deals with the distribution of information. This can be implemented by sending an event to a console, paging or mailing a security officer or any other action that involves notifying the appropriate person. These responses are executed after the attack has been detected by the IDS. Passive alerting is sometimes referred to as passive response.

3.3.3.2 Reactive Response

Reactive responses change the surrounding system environment, either in the host on which the IDS resides or outside in the surrounding network. The main goal of these responses is to stop the attacker from gaining further access to resources, thus mitigating the effects of the attack. Reactive responses are also executed after the attack has been detected by the IDS.

3.3.3.3 Proactive Response

The only difference between proactive and reactive responses is when they are executed. Proactive responses intervene and actively stop an attack from taking place. A proactive response could be to drop a network packet before it has reached its destination, thereby intervening and stopping the actual attack. A reactive response would have been able to terminate the ongoing connection, but it would not have stopped the packet that triggered the IDS from reaching its destination.

3.3.4 Usage Frequency

An IDS can monitor a host or a network either by doing analysis in real-time or by scanning at regular intervals. The most common approach is real-time scanning but tools exist for analyzing log files at regular intervals.

3.3.5 Detection Paradigm

The chain of events that constitutes an attack can be analyzed in two different ways. One way is to look at the state of the system. This paradigm is able to detect if a system is in an error state, e.g. that a non-qualified user has gained access to a system or that a file that should not be changed has been changed. The other way to detect an attack is to recognize the transitions between the different states, that is, when something is actually happening (not after it has happened). In the examples above,

this kind of system would detect the user during the actual login or detect the file change during the actual change (i.e. when the file is written).

Note: In the original taxonomy [4], Debar has divided the detection paradigms even further into two different categories; non-perturbing and proactive analysis. In this thesis only systems performing non-perturbing analysis (that is, observing the system's states or looking for transitions between those) are considered and tools that actively find vulnerabilities by trying to trigger them (proactive evaluation) are not. Such tools exist (usually called vulnerability scanners) but they are more of a complement to intrusion detection (e.g. to find vulnerabilities automatically and then configure the IDS according to the detected information).

4 IDS Today

This chapter covers the different technologies found in commercial IDS products today. What the technologies do and why they are used is explained. The taxonomy is used to classify the technologies, and therefore the chapter layout will follow the taxonomy structure.

Keep in mind that the terminology used by the vendors can differ from the one used here. However, their methods and technologies fit into the descriptions in this chapter. Remember that one method does not exclude another.

The vendors studied are:

- Internet Security Systems
- Cisco
- Enterasys
- Symantec
- NFR Security
- Intrusion.com
- Entercept Security Technologies
- Recourse Technologies

Note: Symantec recently bought Recourse Technologies.

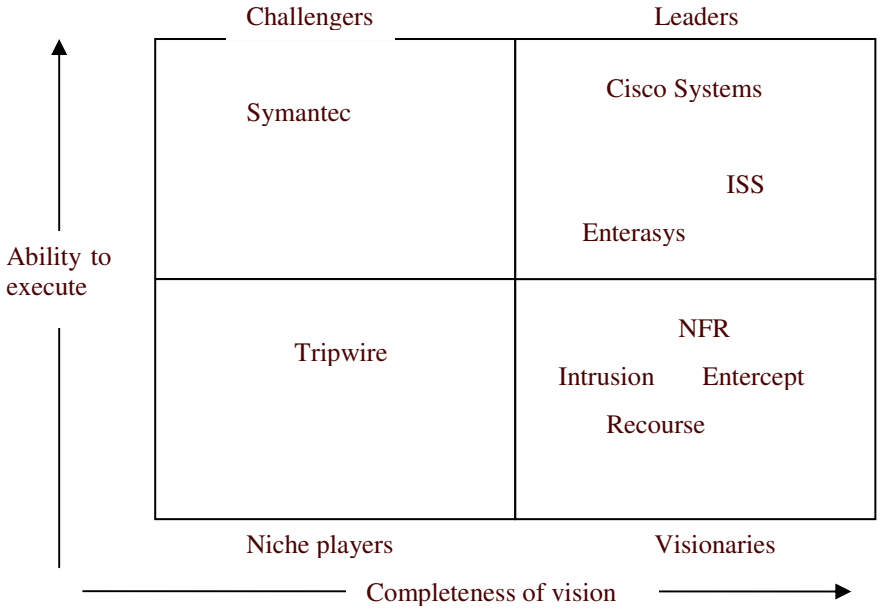

Figure 5: Intrusion detection system 1H02 Magic Quadrant.

These vendors were chosen since they appear in Gartners Intrusion Detection 1H02 Magic Quadrant (see figure 5) [5], which lists the leading intrusion detection vendors. The magic quadrant looks at both the strengths of the individual vendors as well as how technically advanced their products are. Gartner updates this report twice a year and this is the most recent publication available at the time of writing.

Not everything that exists in these products is covered. Most of the information found in this chapter is a combination of white papers, technical information found at the vendors' websites and independent product surveys.

Note: Although Tripwire is present in the Magic Quadrant, it is not covered in detail in this thesis. Tripwire produces file integrity assessment (FIA) products that fall outside the scope of our taxonomy. These tools monitor the state of system and application files, or the registry. They do this by taking an initial "snapshot" of the clean system, usually in the form of cryptographic hashes of the monitored objects. By recalculating new hashes at regular intervals and comparing them with the stored snapshot, the FIA product can detect if any of the monitored objects have been altered. A FIA product does not use any audit source that exists in the taxonomy.

4.1 Audit Source Location

This section explains where most of the IDSs are deployed and how they gather their information. It also covers how the different parts of the system communicate and how alerts are presented to the security officers. The model in chapter 2 is used to expand our image of what an IDS product is.

The products studied can be divided into four different categories based on the type of sensor they use; network-based IDS (NIDS), host-based IDS (HIDS), hybrid IDS and host-based intrusion prevention systems (HIPS). The difference between these types of products lies in where they listen, what they listen to, the way they detect attacks and how they respond. There is also something called a manager that collects the events that the sensors produce. This manager is merely a sensor with events of other IDSs as input. The functionality of the manager will be covered more thoroughly in section 4.1.4.

4.1.1 Network Packets

A network-based sensor does what the name suggests; listens to network traffic at the network layer. These sensors are generally placed at important nodes in the network such as in the DMZ or in different subnets on the internal network (see figure 6). Usually this is a dedicated machine with a network card put in promiscuous mode so that it can listen to all network traffic passing by.

The NIDSs of today usually support 10/100 Mbps, full duplex and fully saturated networks. Gigabit NIDSs exist that supports up to 90% saturation level on gigabit networks, e.g. ISS RealSecure Gigabit Network Sensor and Cisco IDS. Since more and more networks are switched, the network sensor needs to see all the traffic passing the switch it resides on. One solution is to have an IDS integrated into the switch. Another solution is to use a specific port called the span port, which mirrors all the traffic on the switch.

4.1.2 Host and Application Log Files

Host-based sensors reside on single hosts to detect attacks directed at that specific host. Usually they are placed at critical machines such as web servers or firewalls. Most of the host-based sensors are looking at different logs generated by applications and the operating system. Host-based sensors are often seen as a complement to NIDS. Today, the most common sensor seems to be the NIDS.

The main advantage of a HIDS is when an insider attack occurs. An insider attack usually looks like ordinary traffic since it mainly is legitimate traffic coming from a legitimate user, even though the user is up to no good. A HIDS can detect this by looking at strange access times, failed logins and so on. Another advantage with a host-based IDS is that the HIDS has no problem with encrypted traffic since it is not looking at network traffic in any lower layer of the protocol stack.

Hybrid IDS:s exist on the market today (e.g. RealSecure Server Sensor), but they are relatively new. Hybrid IDSs are a combination of NIDS and HIDS. They reside on one host and only detect attacks directed at that host. The main advantage with hybrid IDS is that they have most of the advantages of both network and host-based detection while excluding most of the disadvantages. The sensor still monitors the network traffic and detects attacks in the network layer of the protocol stack. However, the sensor also detects attacks at higher layers and therefore it can detect attacks hidden in encrypted sessions such as IP sec or SSL encryptions. The sensors can also monitor application and OS logs. The big disadvantage of hybrid IDS compared to pure NIDS is that the sensor only detects attacks directed at the residing host. A correctly deployed NIDS can (potentially) detect attacks on the entire network segment on which it resides.

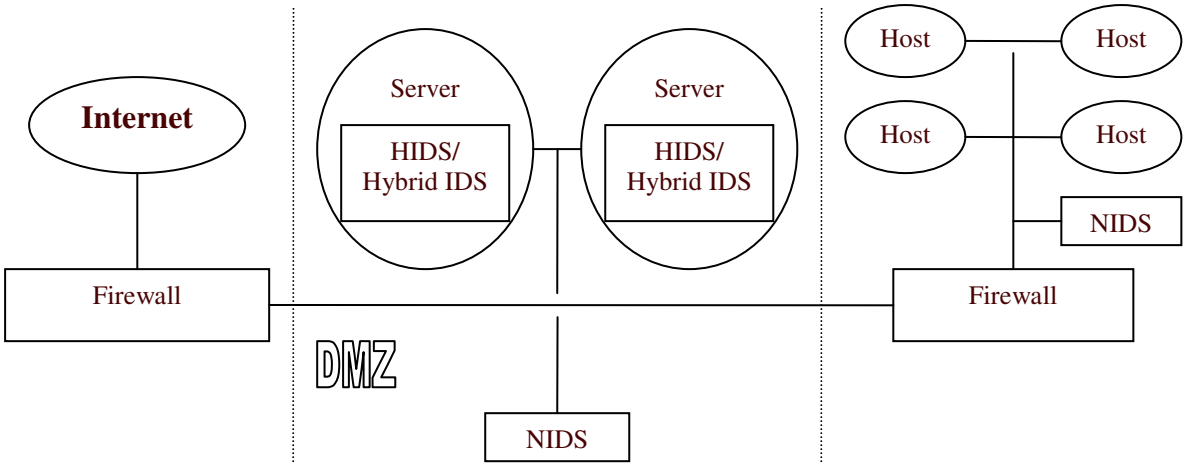

Figure 6: Examples of good places to put the different sensors.

4.1.3 System and API Calls

Another way to detect attacks at the host is to look at system and API calls. The HIDS installs itself right above the kernel of the system, where it intercepts system and API calls. The HIDS understands the context and parameters of the calls and evaluates them to see if they are “good” or “bad”. Since all attacks against a system must be carried out by system calls, this audit source gives the possibility of detecting all

attacks directed at a system. Another advantage with this method is that no encryption is used at this level. Compared with analyzing log files from applications and operating system this method gives the IDS the advantage of seeing everything that happens in a system instead of the things that other products think is important. There is also the possibility that someone has tampered with the log, which is not possible when the IDS looks at system calls.

4.1.4 IDS Sensor Alerts

Most IDS solutions do not include just sensors. Instead, they use something called a *three-tier* deployment strategy. This means that the IDS is made up of three separate layers. The first layer consists (most often) of several sensors (NIDS, HIDS or hybrid IDS) that collect audit data and produce events. These events are sent to the second layer that usually consists of one machine called the manager. The manager takes care of all the events using techniques such as correlation and aggregation (see section 4.2.1.5). The console, which makes up the third layer, is mainly a graphical interface that displays alarms and provides a way to update the sensors' knowledge databases and the policy rules. Even though this layer seems very cosmetic, it is a very important layer. Since there are many alarms for a security officer to keep track of, the presentation of them is very important. By displaying the alarms in a well-structured way, it is easier for the security officer to detect important events. Most IDSs support the use of several consoles.

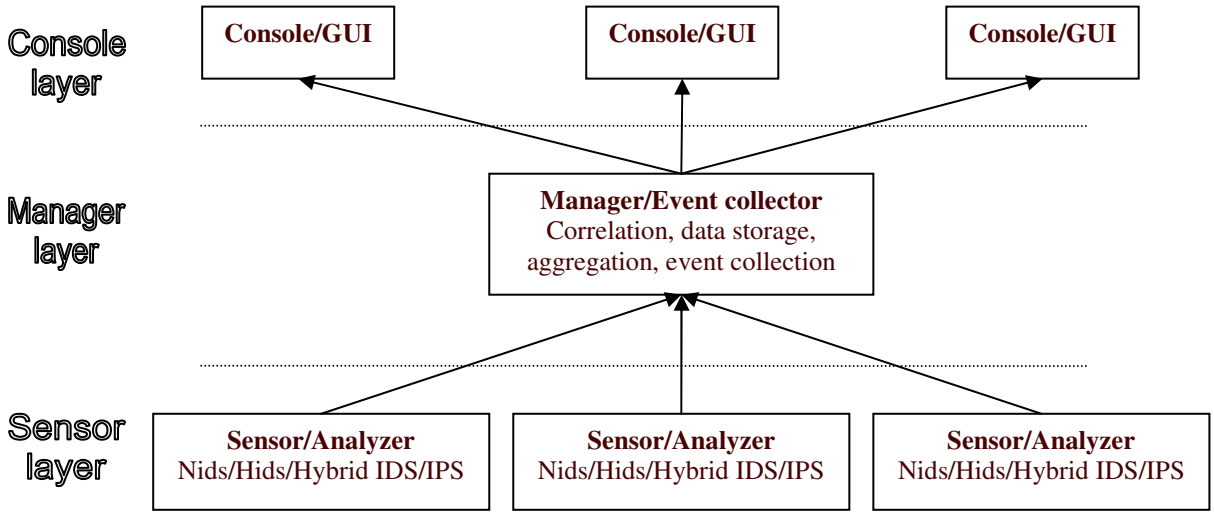

Figure 7: A general three-tier model.

In figure 7 the general layout of the three-tier deployment strategy is presented and in figure 8 the IDS model is used to explain the different layers. Figure 8 shows what components are present at the different layers. Most of the detection work is done at the sensor layer. The sensor gathers information from the audit source and uses the analyzer to classify the data and create events. The events are then forwarded to the response unit, which uses the policy rules to take appropriate action. Apart from any active response taken by the response unit the events are forwarded to the manager layer.

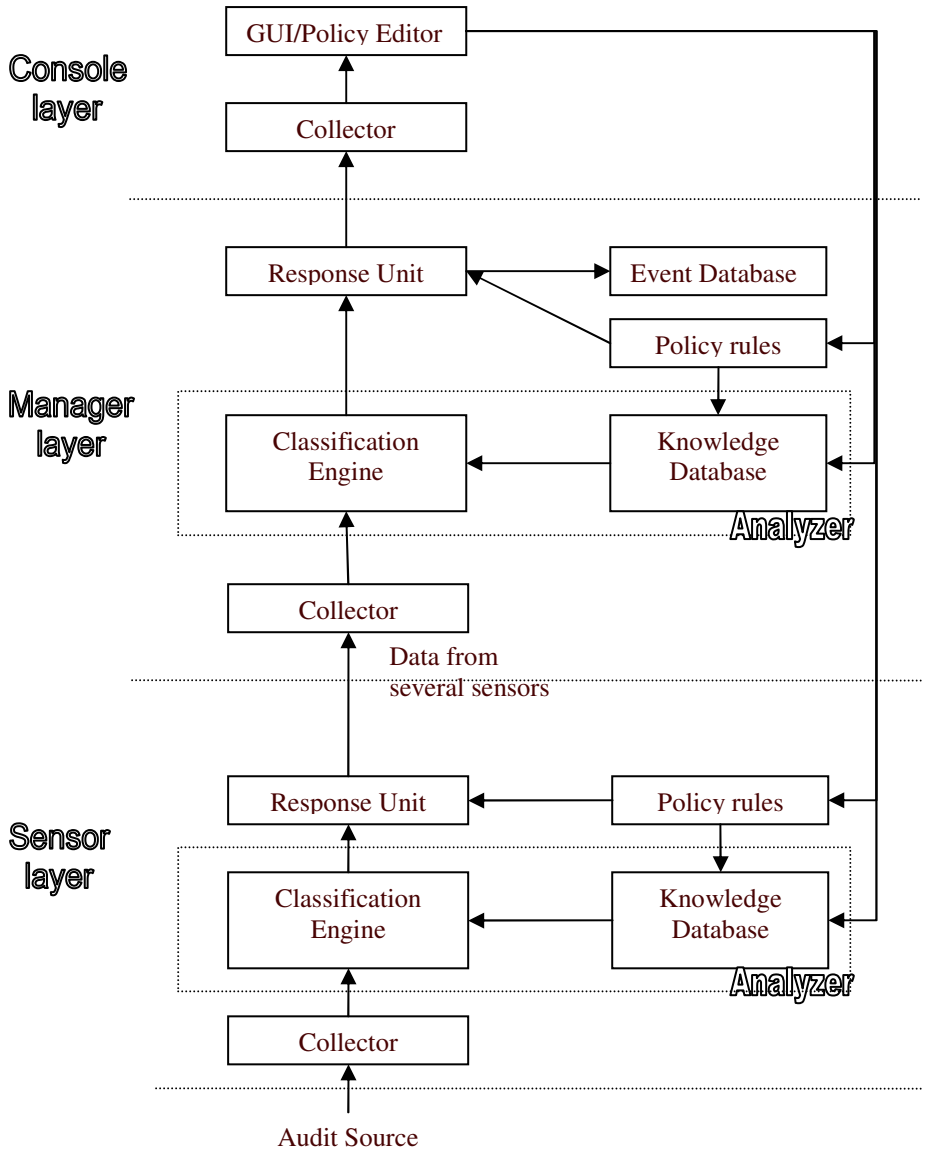

Figure 8: A closer look at the three-tier model.

At the manager layer, a collector gathers events from one or more sensors. The events are forwarded to the manager's response unit, which check its policy rules to know what kind of action to take (if any). One type of action could be to store the events in the event database. Depending on the response, the events may be forwarded to the console layer.

At the console layer, a collector gathers the events sent from the manager layer. These events are presented to a security officer through a graphical user interface (GUI). At this level, the security officer can distribute updates of the knowledge database and policies to every other layer.

Note that there are also systems using a more basic two-tier model, in which the console and management layer is combined into one central console unit.

4.1.4.1 Security Issues

The use of central management and any distribution of events across the network call for a secure way of communicating. The manager must be able to identify and authenticate sensors sending alarms. This is usually solved with some kind of public-key encryption.

In addition, it is important to know if a sensor goes offline, therefore sensors often send heartbeats. Heartbeats are messages sent at regular intervals to let the manager know that the sensor is online. The console and the manager also need to be able to communicate securely. This is solved in the same manner as the sensor – manager communication.

4.1.4.2 Scalability Issues

IDSs using the three-tier model should scale very well. To add a sensor that guards a new subnet or a new host is no problem. As long as the installation process is done correctly the events will be sent to the manager, processed, and then displayed at the consoles.

4.2 Detection Methods

How does the analyzer actually detect what kind of events are suspicious? There are several ways of doing this and the different approaches offer different advantages. Remember that the IDSs can use one or more of the following methods.

4.2.1 Knowledge-based

This section covers knowledge-based detection methods, but also correlation and aggregation. Both correlation and aggregation are something that commonly takes place at the manager layer and not at the sensor layer since an IDS at the sensor layer does not have an overview of the entire network. These methods use some kind of knowledge to group the events and determine the level of seriousness of the events.

When pattern matching, stateful pattern matching and protocol decode are described, the case when the analyzer uses network packets as audit source is used. This is not the whole truth since these methods can be applied to other audit sources as well. Pattern matching can be used for finding specific strings in log files, system calls or IDS events and the idea for protocol decode can be applied to system calls.

4.2.1.1 Pattern Matching

This is the most basic kind of misuse detection for NIDS. Pattern matching systems look for known sequences in e.g. single packets. The strings looked for usually depend on the protocol used and the destination port. A signature could look for a TCP packet destined for port 80 containing the string “XYZ”. If the analyzer finds this packet, misuse has been detected and an event is created.

However, with pattern matching it is easy to evade detection if the attacker knows it is used. An attacker could fragment the payload and send “XY” in one packet and “Z” in the next. The pattern matching detection method does not detect this since it does not remember earlier packets. This is why it is more common to use stateful pattern matching. [1]

4.2.1.2 Stateful Pattern Matching

Stateful pattern matching is similar to pattern matching but in addition the sensor keeps track of earlier packets within one session. This way, fragmented packets will be put together and pattern matching can be applied session-wide, which makes it harder for attackers to evade the IDS. There are still some problems though. This method can be resource exhausting and only a limited number of sessions can be monitored simultaneously. Attackers can exploit this by increasing the time between the fragmented packets to make the IDS drop the session before the entire content has been transmitted. In addition, an IDS using this technique can be drained by an attacker who creates many sessions just to get the resources of the IDS exhausted. Eventually the IDS will be forced to drop some sessions. [1]

4.2.1.3 Protocol Decode

Protocol decode is a more sophisticated way to look at the different packets. With protocol decode the analyzer knows more about what the different parts of a packet actually mean and how they should look. When the analyzer receives a packet of a certain protocol the analyzer interprets what the different bits in the packet represent.

There are two parts to protocol decode. The method can look at the different fields in the packet header and make sure everything is as it should be (according to the protocol specification), e.g. that the flags are used in a correct manner or that the field lengths are as they should be. In addition, using (stateful) pattern matching combined with protocol decode, the signature can specify in what field of the packet the patterns should appear (in the data field as an example).

Further down, behaviour-based detection is described and in some ways protocol decode is behaviour-based. When protocol decode examines if a packet is coherent with the protocol specification, the method knows what is normal and generates events for deviations. This is the exact thing that a behaviour-based method does and just one example of how the different methods blur together.

Usually the products support protocol decode, but merely for knowing what to make of the different parts of packets and not to detect anomalies according to any specification since those usually are ambiguous. After knowing what to make of the different parts of a packet, some kind of pattern matching method is used. [1]

The products studied in this thesis (that has protocol decode in some manner) support a wide variety of protocols. These protocols are not only network protocols such as IP. Protocols all the way up to the application layer are supported including common protocols such as Telnet and HTTP.

4.2.1.4 Heuristic-based Analysis

This is an algorithmic approach for detecting suspicious traffic. As an example, the number of connections from a specific source can be used to detect strange behaviour, e.g. port scans. [1] One algorithm used to do this is the leaking bucket algorithm. This algorithm remembers the number of ports being touched by one source. After a certain amount of time, the algorithm drops connections, but if the total number that is in memory (in the bucket) exceeds some specific value (the bucket gets full) the algorithm generates an alarm. [11] The heuristic-based approach is a good way to

detect these kinds of attacks. The main problem with this approach is that the algorithms usually have to be very fine tuned for every specific network it resides on since networks usually have different behaviours. [1]

4.2.1.5 Aggregation and Correlation

Aggregation and correlation is the key to collect events from several sensors, analyze the complete set of events, and present it in a structured way. Aggregation is used to consolidate events of a specific kind (as an example the IDS can display all the events in groups regarding the source-IP). This way a security officer can easily see if some IP-address is causing a large amount of alarms. The technique is mainly used to identify events that have properties in common and group them together.

Correlation is used to see the connection between different pieces of data. An example is to use vulnerability scanners in a network. These scanners give information about what type of attacks the network is susceptible to. If the IDS correlates this information with the events that it detects, it can remove or at least downgrade the severity of those events that the network is not susceptible to. This will decrease the number of irrelevant events presented to the security officer and thereby increase the chance of detecting dangerous attacks. Another example could be to correlate incoming events with a list of IP-addresses that previously have been the source of real attacks towards our network. This list is then used to adjust the seriousness of all new attacks that originate from any of the addresses on the list.

4.2.2 Behaviour-based

Behaviour-based analysis is also called anomaly detection. As previously described, any method of this kind detects deviations from what the system knows as normal. Anomaly detection has been around for a long time as a concept but it has not grown and evolved enough to take a larger share of the commercial market. There are too many problems around anomaly detection today for it to be used in any larger scale.

4.2.2.1 Statistical Anomaly Detection

Statistical anomaly detection detects changes in behaviour using statistical profiles covering e.g. traffic flow. If the traffic flow deviates from these statistical profiles, an event is generated. This could be a highly inaccurate method and there can be problems when legitimate changes occur in a system environment.

4.2.2.2 Behavioural Rules

Entercept's IDS uses behavioural rules. These rules demand that a system only does what the system is supposed to do. A system usually has several services running. These services are only supposed to do certain things, e.g. a web server should never access the password file of the system since its job is to send out web pages. The behavioural rule for the web server service is that it should only access web pages and, if it accesses anything else, something is wrong and therefore an event is generated.

4.2.2.3 Protocol Anomaly Detection

Vendors often claim to use protocol anomaly detection (e.g. Cisco IDS). This would mean that they detect deviations from what could be considered normal according to

the protocol specification. In other words they use protocol decode as described in section 4.2.1.3.

4.3 Behaviour on Detection

In this section, the different types of responses found in the analyzed systems are presented.

4.3.1 Passive Response

This section will explain how passive responses are used by IDSs for harvesting and propagating information about attacks.

4.3.1.1 Session Recording

Some IDSs can record TCP sessions, e.g. a telnet session, when a detected attack takes place. The session can later be replayed to show, step-by-step, what the attacker did and the damage he caused. The evidence could be used for legal actions, even though this is very rare.

4.3.1.2 Trace

This response executes a trace in order to find the source of the attack. The trace starts at the destination (the attacked target) and follows the flow of traffic back to its origin. This is done by utilizing access control lists and logs of routers and by sampling traffic at key points in the network. As it discovers additional locations in which the flow exists it repeats the process, either until it reaches the origin of the flow or reaches a network that it cannot track further into. This method could find the source of an attack, even if the attacker uses a forged IP-address trough IP-spoofing.

4.3.1.3 Log

This response is often used in combination with other responses. The event that caused this response is logged with additional information. The information can vary but in general it is attack related information like attack type, date, time, source and target of the event. The log is usually saved in a stand-alone event database. This database is a valuable source of information, as will be discussed later in section 4.3.4.

4.3.1.4 Notification

This is the most common response. Notification deals with how to notify the security officer that something is amiss. This can be done in several ways. Primarily by sending an alert to the console, where the event will be displayed according to severity. Examples of other notifications are the generation of an e-mail, sending a SMS or a page request over a dialup connection.

This response method can be used when the IDS wants to pass on information to other IDSs, provided they support the same messaging format.

4.3.1.5 SNMP Trap

Simple network management protocol (SNMP) was originally developed to ease the task of a network manager. It is an asynchronous protocol, which utilizes UDP to communicate with network devices, e.g. routers. There are several commands; *get, set,*

trap, which can be used to get information, alter settings and raise alerts. When used in combination with IDSs only SNMP trap is mentioned. This can be misleading since SNMP traps are originally status messages sent by network devices to network management systems (NMS). When the NMS receives a trap, it may act by sending sets in order to reconfigure a network device. When an IDS sensor sends a trap, some form of NMS must also receive the message. This can be an IDS console or a SNMP listener like IBM Tivoli or HP Openview. It is then up to the NMS to take the appropriate action (e.g. reconfiguring the ACL of a router). To sum up, using SNMP would permit IDSs to react across differing equipment in a coherent fashion in the event when a vendor monoculture is absent.

This response method can be used when we want to pass on information to IDSs from other vendors that do not support the internal messaging format.

4.3.1.6 Spawn Process

This response is hard to categorize since it can do virtually anything. In its simplest form, it could be used to run a batch-file or a program, e.g. an antivirus application. Other possible uses could be to generate an e-mail or shutdown a computer. The possibilities are endless, but note that even though the IDS may start another process, it is not evident that it can incorporate the result.

4.3.1.7 Forged Responses

Some IDSs offer the ability to respond to system scans with spoofed data. Due to the vast amount of script tools as well as vulnerability scanners, it is very easy for an attacker to scan a range of addresses for OS information or vulnerable ports. In answer to this, the IDS may respond to such scans with what appears to be legitimate data, thus confusing the scanning tool into believing that vulnerabilities exist. These “false positives” will hopefully slow the attacker down long enough for the security officer to acknowledge the attack and take the proper countermeasures.

4.3.2 Reactive Response

This section explains how reactive responses are used by IDSs to mitigate the effect of an attack.

4.3.2.1 Blocking

When an attack is detected, this response can block the attacker from further reaching the attacked target. There are several ways of doing this, but at some point it involves the reconfiguration of an ingress point to the network, e.g. a router or firewall. The IDS dynamically updates the routers/firewalls access control list (ACL) in order to deny access for the intruder. Since different routers/firewalls support different protocols for communication, e.g. SNMP or OPSEC, it is not obvious that an IDS can utilize this response even if it claims to do so.

Some of the IDSs studied allows the administrator to specify a range of IP-addresses that will never be blocked, to cope with false alarms.

4.3.2.2 TCP Reset

This response sends a TCP reset to terminate the active TCP session between the attacker and the target. It can be sent either to the attacking source or to the attacked target. The sequence number of the TCP reset packet is of importance since the target/source only accepts packets in sequence.

4.3.2.3 Redirection

Some IDSs offer the ability to redirect the attacker into a honeypot or a honeynet. A honeypot is generally a program or service that mimes the appearance of a computer. The honeypot contains no data or applications critical to the business, but it appears to be the real thing. When an attacker is redirected into the honeypot, every move is recorded. This information can be used to identify who the attacker is, what he is after, what his skill level is, and what tools he uses. The longer the deception of the honeypot being a real computer can be kept, the more knowledge the system gains.

A honeynet is a network of honeypots, which let the attacker move around in a controlled environment.

4.3.2.4 Modify User Accounts

This response type only exists in the host-based IDSs of the products covered. When a certain event triggers this response, the IDS will logout the user whose account caused the event to fire. The IDS may also lock the account preventing the user to login again.

4.3.3 Proactive Response

Among the seven IDSs studied, only Entercept and Intrusion utilizes this kind of response. These systems are host-based intrusion prevention systems that reside just above the OS kernel on their hosts.

The preventive response taken by the IPS terminates an attack action before it can execute. This is followed by some sensible error code being returned to the invoking application so that it is not obvious that an IDS has terminated the action.

4.3.4 Post Processing

This section explains how the information gathered by the IDS can be used in an offline environment. When the IDS has been online for some time, a huge amount of information has been stored in its event database. This information provides a valuable source for learning about the surrounding systems, finding flaws in the network configuration and for generating reports.

4.3.4.1 Forensics

When an intrusion has occurred, it is very important to find out as much as possible in order to prevent it from happening again. Using the information stored in the event database is essential for this. Most IDSs offer some kind of forensic tool. With this, advanced queries to the database can be executed. In the cases where such a tool is non-present, the event database usually supports SQL-queries.

4.3.4.2 Reports

The information stored in the event database can be used to generate a variety of reports. The reporting functions found in the IDSs studied differ greatly and in some cases are non-present. Third-party programs exists (e.g. netForensics, Crystal Reports and FastAnalysis) that can be used in combination with IDSs. The reports are generally statistical summaries of the attacks sorted by some criteria, e.g. most frequent attack types, attacked IP-addresses or login/logout history. Some IDSs even offer the customization of reports although this is quite rare. In general, the built in reporting features of IDSs are quite weak, therefore using an external program for this can be a good solution. The layout of the reports is similar to spreadsheets or basic bar charts.

4.3.4.3 Configuration Flaws

Many false positives can be the result of a badly configured network environment. By using forensic tools or generating reports it is easy to get an overview of the network environment. If many alarms originate from a particular host or network segment the security officer can analyze this area further to find out exactly what causes the alarms. In many cases, a peak in alarms is due to some kind of misconfiguration and by fixing this, the network environment will become more stable and secure.

4.4 Usage Frequency

The usage frequency in the studied products varies but most of them use continuous monitoring. Some products, e.g. NFR HIDS 2.0, can be configured to perform periodic analysis for all or a subset of the signatures.

4.5 Detection Paradigm

This thesis has not been able to determine the detection paradigm of all the products studied. This is something that the vendors do not talk about in their product sheets and the surveys mention nothing about this. Considering IPSs, it is clear that they need to look at transitions rather than states, since they need to detect and stop an attack before it is carried out and an IPS can only do this by detecting the transition and stopping it. If the IDS instead looks at states, it would detect if the state has changed. However, the state only changes after the attack has been carried out. At this point, it is too late to stop the attack, the IDS can only try to stop any further damage from the attack.

5 Problems and Challenges

This chapter identifies the main problems and challenges with the products covered in chapter 4.

5.1 Audit Source Location

Since the audit source is the most important source of information about ongoing activity in the supervised environment, the result of problems here can be devastating for the performance of an IDS. This section covers the main problems with the different audit sources.

5.1.1 Network Packets

The audit source most used today is network packets. Almost all the vendors studied have some kind of network-based sensor. When data is picked up before it has reached its destination, as is done when picking up network packets, there is a problem if the data has been encrypted in any way. If so, the actual payload cannot be seen and the NIDS becomes nearly useless since the content of the packet cannot be examined. Any malicious behaviour hidden within the packet can pass undetected. The only thing the NIDS can do is to look at the header of the packets and see if the traffic pattern looks normal. This means that the technologies that can be used in this case is protocol decode and statistical knowledge about how traffic should look (e.g. volume of network traffic).

Another problem is that the increasing network complexity requires more and more of IDSs. More networks are switched in order to maximize performance as well as logically structure the corporate networks. To successfully monitor a switched network an IDS is either needed on each individual segment or on each of the switches span ports. Either way, it demands more in terms of scalability and aggregation of the IDS in order to function well. Even if most of the IDSs today scale well, aggregation and correlation properties are not satisfactory.

5.1.2 Host and Application Log Files

If an IDS is looking at log files from applications and operating system the IDS has to rely on the application or OS to log the correct data and a problem arises if too little data is logged. An IDS will not be able to detect attacks that do not show up in the log file. This problem is not only appearing when an application or OS logs too little data, it is also evident if there is any possibility for a user to modify the log file. If this is possible an attacker could, potentially, gain access to the log file and remove important events before the IDS discovers it. This makes it easy for attackers to evade detection by the IDS.

A performance problem can arise if we want to increase the amount of logging. Logging is resource exhausting and in some cases the system cannot cope with the heavy load of logging all data.

5.1.3 System and API Calls

A HIPS processing system and API calls is usually installed just above the kernel in the host. By processing low-level calls, it is easier to determine if something is wrong since the plethora of legitimate calls at this abstraction level is fewer than on a higher level. The downside of it is that several attacks exploit the same flaw and therefore look the same on this level in the system. Therefore, it is very hard to differentiate between different attacks and to truly know the originating process. Because of this, an IDS that uses system and API calls as its primary source will sometimes generate the same general event for several different attacks. This makes finding and stopping the source of the attack much harder.

5.1.4 IDS Sensor Alerts

The interoperability between IDSs of different brands is very poor today. All the studied products send encrypted internal messages between the different parts of the IDSs for propagation of event data. The problem is that they all have implemented the communication different from each other. This makes it very hard to manage and supervise several IDSs from different vendors at the same console.

The SNMP trap response is an attempt to offer a standardized way for communication. Unfortunately, the underlying protocol (UDP) is not the best for such an important task since it is unreliable. In addition, SNMP messages are quite simple in nature and limit the amount and type of information that can be sent.

5.2 Detection Method

One problem that all detection methods suffer from is the difficulty of keeping the false positive rate small relative the true positive rate. Even if the IDS has a small false positive rate, say 0.1%, the number of false alarms related to the number of real alarms will be high. This is true if the actual number of true positives in a network is small related to the traffic volume. An example, the IDS above has a false positive rate of 0.1%. Lets say that the number of units that can trigger an event is 1 000 000 each day. Among these units, 1000 false positives will be found. According to [13], the approximate number of real alarms is one or two per 1 000 000, let us say two for this example. Therefore, if the IDS finds 100% of the true positives, the security officer will have to find those two true positives among 1002 alarms. This implies that the false positive rate needs to be extremely low in order to get the number of false alarms small related to the number of real alarms. The problem is based on a statistical phenomenon called the base-rate fallacy. [13]

5.2.1 Knowledge-based

Knowledge-based detection methods are very good at detecting specific attacks and they give good information about what kind of events they have detected. The main problem with knowledge-based detection methods is that they cannot detect attacks for which no signature exists. Without constant updating of the signatures the IDS will have problems detecting new attacks.

Another problem with signatures is that if they are too narrow an attacker could potentially bypass the IDS by changing the attack slightly, maybe by just adding

irrelevant data. If the signatures instead are too generic, there may be problems with too many false alarms.

Due to the increasing speed of computer networks, stateful pattern matching and heuristic analysis will require more resources of the IDS. If the throughput increases, the number of open sessions and fragmented packets increase as well. This will force the IDS to be able to keep more sessions alive and store more fragments simultaneously, thereby demanding more in terms of hardware.

5.2.2 Behaviour-based

Behaviour-based detection methods have knowledge about how something should look instead of how it should not look. This is a very good approach to detecting attacks since it can also detect new attacks, but these methods have some major disadvantages. One big problem is how to define what is normal. Looking at, for example, network packets to detect anomalies (as the behaviour-based part of protocol decode does) the IDS searches for deviations from the protocol specification for the actual protocols analyzed. But usually the different implementations of protocols vary a lot. Therefore, the IDS has a very vague knowledge about how a packet should look, and because of this, the possibility of missing attacks increases. This is an exact opposite of the problem with signatures. If what is normal is too widely specified, attacks can be missed, and by specifying too narrowly, the result could be too many false positives.

Another problem with behaviour-based detection methods is that they cannot specify details around an alert as well as knowledge-based methods. When a knowledge-based method triggers an alert, the signature that detected the alert is usually associated with some kind of attack information. Therefore, a knowledge-based detection method better specifies what triggered an alert, e.g. a Code Red attack. A behaviour-based method can only tell what is not normal, e.g. a statistical anomaly detection method that looks at the network traffic flow can only say that the traffic is unusually high or low. This demands more from the security officers that are supposed to resolve alerts. In addition, it can be hard to specify any kind of active response to an alert like this, since we do not know the cause of the alert.

5.3 Behaviour on Detection

Although much functionality has been poured into IDSs in order to make them as self-functioning as possible, the need for human interaction is still important. Humans have the rare property of sensing that something is not right, detecting patterns where no obvious patterns exist. This “fuzzy logic” is what separates man from machine and it is the most important property that we try to mimic in computer systems. Since artificial intelligence is fairly undeveloped in commercial intrusion detection systems, we still need humans.

Automated responses are a result of trying to make IDSs more independent of human interaction. Although it might be smart to automate some sequences, since humans can never react as quickly as a machine, it also introduces a great risk. When used with proactive/reactive responses, an IDS could easily be fooled into blocking random addresses. Carelessly implementing automated passive alerts could result in the

security officer being paged in the middle of the night every time someone scanned the ports on the outer firewall.

5.3.1 Passive Alerting

The most common response used in the studied IDSs is notify console. Every time an event triggers this response, the user interface of the IDS is updated with the new alert information. The monitoring security officers review the alerts, take the appropriate actions and remove the alerts after they are treated. As discussed in section 5.2, the base-rate fallacy results in an overwhelming amount of alerts that have to be reviewed every day in order to find the few true positives. This increases the cost of ownership for the IDS dramatically since additional staff is needed.

To make the most of the human-computer interaction and to minimize the time needed to decide whether an alert is benign or hostile, good user interfaces are needed. This is an area in which IDSs could use some improvement. IDSs were originally strictly text-based, but with the introduction into the commercial world, a more user-friendly approach was applied. The ability to quickly drill-down an event chain to see what actually triggered an event, as well as doing extensive searches to cross-check causes and finding patterns is of utmost importance in an IDS. It should be possible to access everything from generated events to recorded sessions, all from the same user interface. Unfortunately, this does not seem to be the case in many of the studied IDSs.

Passive alerting deals with the distribution of information, which often includes storing it on disk. Specifically session recording and the logging of events will demand more in terms of storage with the increasing network speeds. True session recording, where the packet payload is recorded as well, requires even more resources.

5.3.2 Reactive Response

One problem with these responses is that even if the attacker is denied future access to the systems, the attack that triggered the response went through. If the attack involved installing a listening service, like a telnet daemon, blocking the IP-address of the attack only bought the time it takes for the attacker to hijack a different IP-address and logon to the newly installed service. It is also very dangerous to use responses like this automated since they can be provoked to block legitimate IP-addresses, e.g. your critical business partners.

5.3.3 Proactive Response

False positives are particularly dangerous in this area. Proactive responses prevent system calls from executing, and false positives could lead to a denial of service situation for a legitimate application or process running at the host.

Automatic proactive/reactive responses should only be used when confidentiality and integrity outweighs availability.

5.4 Usage Frequency

Most of the systems studied continuously monitor their audit sources. To be able to detect an attack as fast as possible, this is necessary. The problem is that it can be really resource exhausting. A NIDS monitoring a gigabit connection can have a lot of

traffic to monitor if the connection is heavily loaded. This demands a lot from the NIDS but should not have any real impact on the network performance if the NIDS only listens to traffic.

The severity of the problem increases when a HIDS is used. If a HIDS demands too much resource from its host, the performance of the critical server that the HIDS resides on can degrade. This could mean that a HIDS placed at such a server cannot monitor the system continuously but instead needs to analyze the system periodically to detect if anything malicious has happened.

If periodic analysis is used, the IDS cannot be relied on to detect an attack right away. Attacks can only be detected at the time of scan, and at that time, the attack has probably already been carried out. The choice between continuous monitoring and periodic analysis is a trade-off between system performance and the importance of knowing significant events as fast as possible.

5.5 Detection Paradigm

The problem with detection paradigms is similar to those of usage frequency. By using a transition-based IDS an attack can be detected as it is taking place, similar to continuous monitoring. State-based IDSs detect attacks after they have changed the state of the system, similar to periodic analysis. It is more of a choice than a problem.

6 Result of the Interviews

This chapter covers the result of the interviews carried out with the following Swedish banks:

- Handelsbanken
- Föreningssparbanken
- SEB
- Östgöta Enskilda Bank (Danske Bank)
- Ikanobanken

The result presented is a summary and single banks are not identified due to confidentiality reasons. The summary aims at describing the problems that the banks have identified in today's IDS products, and to some extent how the products are put to use.

The people interviewed were mainly in a leading position and most often from an IT-security department. Those who did not work directly with IT-security had good insight in this area.

The threats that the banks had identified were of both external and internal nature but most of them feared external threats the most.

6.1 The Use of IDS Today

Almost all of the banks use some kind of IDS today, though the usage of the system varies a lot. The majority use a commercial product but in-house developed IDSs were also seen. The use of IDSs is relatively new in most of the banks although one bank has actually had some kind of IDS for almost six years. The one bank that did not use an IDS had performed intrusion tests in collaboration with an external contractor, with an IDS up and running during a two months period. The tests showed that the IDS did not offer any additional security over the other systems. All of the banks that use IDS, except one, have the system maintained by their own employees. The one that has outsourced their IDS is looking for ways to implement a system that is operated by their own staff instead.

The banks that use IDS treat it as a separate system. It does not communicate, in any way, with other systems (e.g. firewalls and vulnerability scanners). Some has continuous monitoring of the system and some only look at statistics and log files in the case of an incident. The configuration of the systems also varies a lot. In some cases, the security department wants to see everything that goes on in the network, and thereby they have quite a lot of alarms and false alarms. The reason for this is that the main use of the IDS is to log whatever happens and with this log create statistics of the traffic flow and the attack patterns. Others have a very strict configuration, where they only see the most important events. Therefore they do not have a big problem with false alarms. The banks that implement this strict configuration also have higher confidence in the IDS and the alerts it generates.

None of the banks see the ongoing maintenance of the systems as a large problem, but the installation and initial configuration was a lot of work. As long as there are no big changes in the network architecture, the only real work, except for handling alarms, was to keep the system updated. This was not a problem for any of the banks.

All of the banks were using some kind of statistical, regularly generated, report from the system, or were soon about to implement this. These reports were one of the main advantages of the system since they give a lot of information about traffic flow, misconfigurations in the network and attack patterns.

None of the banks that had implemented an IDS were dissatisfied with the system. They all thought it was a good complement to their other security systems.

6.2 Identified Problems and the Future of IDS

All of the banks recognized false alarms as one of the biggest problems with the IDSs of today. The confidence in the alarms is not high enough to use automatic responses to any alarms at this stage. Some consider it impossible to use any kind of automatic response ever, since the availability of the system always outweighs the integrity and confidentiality regardless of future improvements. In this case, any decisions in response to alarms have to be made by a human being. Others though, want to see a more trustworthy system so that the use of automatic responses can be implemented.

Another problem identified is the dependence on competent staff members. In order to fully utilize an IDS system, a lot of knowledge about the underlying network architecture as well as the IDS itself is needed. Since both the IDS system and the network architecture vary from company to company, recruiting new staff members is a lengthy and costly process.

One bank wanted the possibility of knowing exactly what the different signatures in the IDS triggered on. This feature is very rare in today's IDSs. Some issues with sensor – console communication were also raised, since some IDSs had problems guaranteeing that this communication did not open a path through the firewall.

A majority of the banks wanted to see more use of anomaly detection in the future. The potential of this technology is great if the problems (see chapter 5) can be solved. A wish for more integration with other security related products was also expressed, mainly to correlate information from firewalls, vulnerability scanners, etc, but also to be able to manage the systems from a single console.

6.3 Conclusion

Today, most of the banks use some kind of IDS in their network. The use of the system varies but none of the banks use automated responses and all of them have problems with the confidence in the events that are generated. The IDSs today are only a complement to existing security solutions, but the usage of IDSs and the investment in these systems will rise in the near future. The overall image that the banks gave during these interviews is mainly positive; the IDSs provide increased security for their networks.

7 Recent Research Advances

This chapter covers recent research advances within the area of IDS. The information presented below is based on scientific papers found in [3]. This book was published in connection with RAID 2002, an IDS symposium. In relation to this symposium, IDS researchers around the world submitted research papers for review. A jury consisting of IDS scientists selected 20 of these papers to constitute the book. The selection process suggests that the papers chosen provide interesting research results and give a fair description of the present direction of IDS research.

Since the research today is not only concerned with issues identified in this thesis, the division used here is different from that of the taxonomy. Not all of the information found below can be related to the problems identified in chapters 5 and 6, but it is still relevant information since it indicates were the IDS community concentrate their research efforts today. This in turn may predict the future direction of commercial IDSs.

7.1 Detection Methods

The majority of the papers in [3] concern detection methods. The biggest weakness identified in this thesis is the one concerning false positives. This problem is directly addressed in the following sections, since the ability to accurately detect attacks is heavily dependent on the success of the detection method.

7.1.1 Knowledge-based

The papers concerning knowledge-based detection methods propose new ways of detecting specific intrusions. Methods for detecting when a computer is being used as a stepping-stone and for detecting buffer overflow attacks are proposed. New approaches to correlation are also presented.

7.1.1.1 Stepping-stone Detection

A stepping-stone is a computer used in a connection chain to separate the attacker from the target. This is done to make the identification of the attacker more difficult. A computer that is used as a stepping-stone may not have been attacked directly, but since it is used with malicious intent, a strong enough reason exists to detect and stop this kind of behaviour.

Detecting Long Connection Chains of Interactive Terminal Sessions and *Multiscale Stepping-Stone Detection: Detecting Pairs of Jittered Interactive Streams by Exploiting Maximum Tolerable Delay* cover stepping-stone detection, but they take completely different approaches to the subject.

The first paper is directed at host-based IDSs and it identifies a way to detect when a specific computer is being used as a stepping-stone. This is done by looking at delays in terminal session communication. When specific packets arrive with long delays between them, there are reasons to believe that the computer has a long connection of computers downstream. Since there is usually no legitimate reason to have a long connection chain of computers, this can be evidence of the computer being used as a stepping-stone in an ongoing attack.

The second paper approaches the problem of detecting stepping-stone behaviour in the network layer. This is done by looking at a network and comparing incoming sessions with outgoing sessions. If the method finds sessions with extreme similarities there is reason to believe that one or more computers in the network is part of a stepping-stone chain.

This method will detect when someone outside the network is using a computer in the network to connect to another computer outside the network. Since this is no long chain, this could be considered as legitimate traffic and, if so, the method will raise a false alarm. The advantage with this approach compared to the first is that the security officer does not need control of the computers in the network. An ISP, for example, has no control of the computers in its network. In this case, the second approach has to be used.

7.1.1.2 Buffer Overflow Attacks

Accurate Buffer Overflow Detection via Abstract Payload Execution covers new approaches to discover buffer overflow attacks. Buffer overflow attacks are specifically vicious since they offer a way to gain increased privileges on a computer. When using a buffer overflow attack, an attacker tries to get the target computer to run the attacker's (malicious) code by exploiting buffer weaknesses. If this is done correctly the attack will overwrite parts of the stack, and potentially the return address, thereby getting the program counter to jump to the attacker's own code. Since the attacker usually does not know the exact position in the buffer memory where his code segment is placed, he can add a so-called sledge to improve his chances of success. A sledge is a long segment of executable code that does nothing except moving the program counter forward towards the malicious code. By looking for long executable code segments, this method tries to identify buffer overflow attacks.

Since this method is knowledge-based, it does not have the main problems of behaviour-based detection, i.e. defining what is normal. Even so it also evades one of the biggest problems with knowledge-based detection, i.e. the need for continuous updates. However, as always there are problems. An attacker could evade detection by dividing the executable code into smaller segments, without reducing the success rate of the attack. The authors of the paper are going to address this problem in their future work and hopefully solve this.

7.1.1.3 Correlation

Two papers discuss alarm correlation and the importance of minimizing the amount of false positives produced by IDSs.

The first, *A Mission-Impact-Based Approach to INFOSEC Alarm Correlation*, has developed a working prototype called the *M-Correlator.* This prototype acts as a manager, receiving alerts from several INFOSEC devices (e.g. firewalls, IDSs, virus scanners and integrity scanners). A relevance score is assigned to the alerts through a comparison of detailed information about the attack target's system environment, known network topology and the vulnerability requirements of the incident type. The vulnerability requirements are provided to M-Correlator by an *Incident Handling Fact Base,* which contains more than 1000 intrusion report types from some of the leading

IDS products. These intrusion reports contain information about what kind of system environment (e.g. OS and services installed at the target) an attack needs to be successful. Next, a priority calculation is performed for each alert based on a user-specified interest level and depending on if critical assets have been targeted. Last, an overall incident rank is assigned to each alert, which brings together the relevance score and the priority of the alert with a calculated probability of success for the attack.

As stated in both chapters 5 and 6, the problem with IDSs producing vast amounts of non-relevant information, as well as false positives, is one of the biggest faced by IDS developers today. Even if some (e.g. RealSecure) use correlation, they only correlate alerts with information from vulnerability scanners. The M-Correlator on the other hand, can be configured to understand which parts of the network that are critical and which attacks that are most dangerous according to the company security policy. By adding this knowledge, the amount of non-relevant attack information can be kept at a minimum.

The second paper, *M2D2: A Formal Data Model for IDS Alert Correlation*, offers a different approach to alert correlation. The authors propose a formally defined model to address the shortcomings of today's correlation techniques. M2D2 integrates four types of information: information related to the characteristics of the monitored information system (e.g. the network environment), information about vulnerabilities, information about the security tools used for monitoring (IDSs, VA-tools), and information about the events and alerts observed. Every part of the model is formally defined in a symbolic notation, and to some extent, implemented through a relational database. It is possible to create functions to find harmful alerts, or detecting false positives, in the proposed notation.

Commercial IDS companies also develop their own models, but most often their models lack the ability for the administrator to easily create new relations for correlation. This may render their IDSs unusable in a multi-vendor environment when new detection and response methods become available. M2D2 offers more than any existing model and by adopting it, the IDS community can work together towards creating better correlation patterns.

7.1.2 Behaviour-based

Three papers cover behaviour-based detection. Two of them suggest new approaches to behaviour-based detection while the third presents some issues concerning blind spots in behaviour-based IDSs.

7.1.2.1 Detecting Malicious Software

Detecting Malicious Software by Monitoring Anomalous Windows Registry Accesses presents a new approach for detecting malicious usage of the Windows registry. In this approach the authors examine how different programs use the Windows registry. By observing the usage of registry keys by both legitimate and malicious programs, the authors have concluded that malicious usage of keys can be detected.

The advantage of this method is that of behaviour-based detection in general, new malicious software can be detected without updates of the IDS. In addition, the identification of training data is not as difficult as in other behaviour-based detection

methods. A clean computer with all the desired applications added can be used to train the IDS. Since the training process is demanding, the method is most useful in a corporate standardized environment where many users have the same programs installed on their computers.

7.1.2.2 Access Control

In *Introducing Reference Flow Control for Detecting Intrusion Symptoms at the OS Level* the authors suggest a way of detecting policy violations by using extended access control mechanisms. The proposed method identifies different domains with different access rights. The purpose of the domains is to identify the operations that can be combined. As an example, if a password change is initiated, the associated domain would permit the operation to change the password file, as long as this operation is executing within the same domain. Within this domain it should not be possible to, for example, read the password file. By creating different domains the proposed method can detect, and potentially prevent, the possibility of exploiting the result of attacks such as the buffer overflow attack.

This method defines what is normal, and therefore it is behaviour-based. One problem with these detection methods is to specify what is normal and what is not. Since specifying boundaries for different domains is fairly easy, this method should not suffer from this problem.

7.1.2.3 Blind Spots

Undermining an Anomaly-Based Intrusion Detection System Using Common Exploits tries to identify weaknesses with behaviour-based detection methods. The authors use a specific behaviour-based IDS and tries to find its weak spots. They succeed in this by locating the blind spots of the program and then altering existing attacks so that the IDS cannot detect them (although it would have detected them before they were altered).

The paper unravels possible ways of evading detection and although it is very specific in what kind of IDS they evade, the authors make an interesting point. They prove that there are weaknesses not yet considered with behaviour-based detection.

7.2 Learning New Attacks

In *Learning Unknown Attacks – A Start* the authors create a server cluster that can resist attacks as well as learn about new attacks after a successful attack has been carried out. The cluster consists of at least four servers. Two of these servers are used to ensure continuous operation although a successful attack has been carried out (these servers can be more than two, resulting in increased fault tolerance). One server takes care of communication with all other servers. No communication occurs without passing this controller. The controller behaves similar to a firewall and an IDS. It can deny an attacker further access and block known attacks. The final server is called the sandbox. In case of a successful attack this sandbox analyzes the events logged by the attacked server. The analysis creates rules that will block further attempts to use the same attack.

This report is very interesting since the authors have succeeded in blocking new attacks without human interference. Even though the servers are susceptible to new attacks at a first stage, the sandbox together with the backup servers ensures continuous operation. The approach used by the authors solves the problems with new attacks, but the solution is too cost intensive to be realistic (as the authors clearly state). This is merely a small step towards creating a fault tolerant system.

7.3 Attack Patterns

Three papers, *Analyzing Intensive Intrusion Alerts via Correlation*, *A Stochastic Model for Intrusions* and *Attacks against Computer Network: Formal Grammar-Based Framework and Simulation Tool* try to establish models for attacks or attack patterns. These models can be used to see patterns in attacks and increase the understanding of attacks and vulnerabilities. The papers do not directly concern the problems in chapter 5 or 6 but they approach the problem of relating attacks to each other as well as increasing the understanding of attack behaviour.

7.4 Testing IDSs

Evaluation of the Diagnostic Capabilities of Commercial Intrusion Detection Systems and *Capacity Verification for High Speed Network Intrusion Detection Systems* concern the use of inadequate IDS tests. The first paper treats the problem of testing the quality of the diagnostic properties provided by IDSs. First it examines existing IDS tests, and after illustrating their weaknesses, introduces a test bed of its own. Four NIDSs are tested, with Snort as a reference, and the result shows that the used test successfully isolated several weak properties that should be improved. The second paper concerns the flaws in today's NIDS tests. Apparently, the majority of tests resemble those used with other networking equipment, such as switches and routers. Since switches and routers do not perform the same level of deep packet inspection, nor require the higher-level protocol awareness, these tests do not provide trustworthy results. Therefore, the paper aims at designing a new set of tests that is specific for stressing the weaknesses of NIDS.

Since reviewing IDS tests is clearly outside the scope of this thesis, this issue has not been covered earlier. Even so, the absence of solid IDS tests decreases the credibility of, and the confidence in, IDSs. Therefore, working towards a unified test bed is desirable for IDS to be better positioned as well-established security products.

7.5 Performance Issues

Performance Adaptation in Real-Time Intrusion Detection Systems, concerns the problem of decreasing detection coverage in high-speed networks. As network speed increases, the resources of an IDS become exhausted and eventually the IDS starts skipping events leading to missed attacks. A way to better control this problem is to let the IDS reconfigure itself during workload peaks so that it prioritizes certain attacks and intentionally skips others. The authors of this paper propose a model that measures the performance cost of detecting an attack and the vulnerability cost of that attack (e.g. a port-scan can be considered less lethal than a DoS-attack, and therefore has a lower vulnerability cost). The resulting optimization problem, minimizing the vulnerability

cost while maximizing the detection performance, is what controls the reconfiguration of the IDS.

Performance adaptation is a highly desired functionality for IDSs to adopt. Even though this thesis has treated speed as a matter of no concern (speed could be related to insufficient hardware instead of software), the performance gap seen in IDSs today may not be easily overcome in the near future. By implementing the performance adaptation functions described here, the problem with lagging IDSs can be reduced.

7.6 Benefits of NIDS

In *The Effect of Identifying Vulnerabilities and Patching Software on the Utility of Network Intrusion Detection* the authors investigate the actual benefit of using NIDS. The authors have looked at three different server software packages, namely Bind Domain Name Server, Apache Web Server and Internet Information Server (IIS). They have investigated how often high severity, remote-to-local attacks have appeared between 1996 and 2002, and how fast updates to these attacks have become available as well as how fast signatures for Snort and a vulnerability scanner called Nessus was available. The investigation shows that software patches to prevent vulnerabilities from being exploited are available before or simultaneously as signatures. In addition, the authors found that these vulnerabilities were infrequently discovered, at most six times a year. After discovering this information, the authors draw the conclusion that NIDS are not that useful for small sites with a limited number of critical hosts. At these sites it is more cost-effective to deploy patches as fast as they become available. In larger sites, with 100's or 1000's of hosts and servers, it is very expensive and practically impossible to maintain software patches on all systems. These sites can clearly benefit from NIDSs since they will detect the attacks trying to exploit the vulnerabilities that have not been patched.

The problem with updating signatures as described in chapter 5 falls into a different perspective with this information. The problem is now not only that of having to update the IDSs, there is also the problem with vendors that do not release signature updates fast enough. As stated in this paper, an alternative to using IDSs is to constantly install software patches when they are released. As the authors note, this is not always easy in a large system environment since patches can create other problems (e.g. compatibility problems). Updating an IDS with new signatures is not a problem compatibility wise. Therefore, maintaining an IDS can be more effective than constantly patching all installed software.

7.7 Conclusions

A large portion of the papers concern detection methods. Prioritized areas are the correlation of security relevant events and finding better and more accurate detection methods. These areas address the problem of false positives and try to reduce the amount of non-relevant information produced by IDSs.

Another area of interest is the building of solid test beds for IDSs. This area seems to have been left behind until now. Good test beds need to be developed in order to make fair comparisons between different IDS products.

Both the papers that address the problem of false positives and the papers covering IDS test beds take steps towards raising the credibility of, and confidence in, IDSs. False positives are the problem that hurts the credibility of IDSs the most since many seem to feel that the alerts generated by an IDS cannot be trusted. If a common test bed for IDSs can be developed, interested parties can easily compare systems and their accuracy. If a recognized test bed can show good results for IDSs, the general confidence in IDSs would rise.

8 Conclusions and Recommendations

This chapter summarises the thesis and covers conclusions and recommendations. Future work is included as a recommendation of work related to this thesis.

8.1 Summary

Intrusion detection systems have gained more and more attention during the last couple of years. Today, IDSs are almost a necessary part of network security in larger corporate networks. With the increased usage, the weaknesses of IDSs have become more apparent. This thesis has presented a model and taxonomy for IDSs, described the different technologies used in intrusion detection and identified problems with these technologies. Opinions of security departments at Swedish banks concerning the usability and future of IDS have also been presented. Finally, recent research results have been covered and discussed.

8.2 Conclusions

The most serious problem faced by the IDS community is the problem with false positives. Chapter 5 identifies this problem and in chapter 6 the banks express the same concerns. The number of false positives in today's IDSs could cripple the future expansion of IDS usage.

Reactive responses have given rise to much controversy. As shown in chapter 6, the confidence in IDSs to make important decisions is generally low. When the outcome of a response and the success of the triggering attack are uncertain, the credibility of the response is seriously doubted. Add false positives to the mix and suddenly the rate of successfully blocked attacks is similar to a random number generator.

8.3 Recommendations

The solution to the false positive problem is threefold. First, better and more accurate detection methods are needed. This includes better and more specific signatures. Second, the need for a communication standard between security systems is evident. If communication could take place in a standardized way between firewalls, vulnerability scanners, intrusion detection systems and a joint manager, the amount of information available to the manager as well as the security officer would be greatly increased. Third, this information could be used to develop a better correlation model. Good correlation of the information from the different systems could greatly reduce the number of false positives. As seen in chapter 7, new and/or improved detection methods as well as models for correlation is a prioritised area for research. Communication is not an issue studied in chapter 7, but steps are taken towards an RFC for a communication language called IDMEF. IDMEF has not been covered in this thesis. For more information, see [13].

What can be done to solve the problem with reactive responses? One approach could be to focus on intrusion prevention systems. As discussed in this thesis, IPSs use proactive responses and by doing so they have better control over what takes place in the monitored environment. By only activating those signatures mature and accurate enough, an IPS could easily provide the same functionality as a firewall, but with an

additional layer of security added. A possible development of the commercial IDSs would be that HIDS are replaced by HIPS, and NIDS and firewalls by a next-generation NIPS, combining both technologies. However, the functionality of IPSs will not be fully used as long as the confidence in the alerts generated remains low. This confidence will only rise if the problem with false positives is solved.

The biggest problems faced by the IDS community have been identified in this thesis. Even though all of them are important, the two discussed in this chapter are believed to have the greatest impact on the future of intrusion detection. In addition, many of the other weaknesses directly originate from the problem with false positives. The future of IDSs depends on how the IDS community address these challenges. Even if no major breakthroughs have been seen in the recent research, the results are steadily improving important areas within intrusion detection. The future will hold even better and more reliable systems provided that researchers continue to address the correct issues.

8.4 Future Work

This thesis has covered several different vendors and their products, but the products have not been studied in any practical way. Performing practical experiments with the products and comparing the result of such a study with the results presented here would be very interesting.

Since this thesis has primarily focused on commercial IDSs, conducting a similar study of research projects and IDS prototypes could prove very rewarding. More information about the internal functionality of IDSs is generally available in the research community. This would lead to a deeper understanding of IDSs and help identifying more weaknesses as well as improvements.

In order to reduce false positives and improve usefulness, better integration and correlation is needed. Products addressing some of these issues exist today, e.g. Tivoli Risk Manager and Computer Associates' eTrust. A possible future study could be to compare IDS implementations in similar businesses, with and without these products, in an attempt to identify the impact on false positives.

9 Bibliography

This chapter contains material referred to throughout the thesis and links to the IDS vendors.

9.1 References

1. **The Science of Intrusion Detection System Attack Identification**, Cisco, http://www.cisco.com/en/US/products/sw/secursw/ps2113/products_white_paper09186a0080092334.shtml, January 2003
2. **The Base Rate Fallacy and its Implications for the Difficulty of Intrusion Detection**, Stefan Axelsson 1999, Chalmers, http://www.ce.chalmers.se/staff/sax/difficulty.pdf, January 2003
3. **Recent Advances in Intrusion Detection,** Andreas Wespi et al 2002, Springer Verlag, ISBN 3-540-00020-8
4. **A Revised Taxonomy for Intrusion Detection**, Hervé Debar et al 1999, IBM Research, http://www.securitytechnet.com/resources/rsc-center/vendor-wp/ibm/ibm-survey.ps, January 2003
5. **Intrusion Detection System 1H02 Magic Quadrant**, M. Easley et al 2002, Gartner
6. **Intrusion Detection**, Rebecca G. Bace 1999, 1st Edition, Pearson Higher Education, ISBN 1-578-70185-6
7. **Computer Security**, Dieter Gollman 1999, 1st Edition, John Wiley & Son Ltd, ISBN 0-471-97844-2
8. **Intrusion Detection: An Introduction to Internet Surveillance, Correlation, Trace Back, and Response**, Edward G. Amoroso 1999, 1st Edition, Intrusion.Net Books, ISBN 0-966-67007-8
9. **Using internal sensors for computer intrusion detection**, Diego Zamboni 2001, Purdue University, https://www.cerias.purdue.edu/techreports-ssl/ public/2001-42.pdf, January 2003
10. **Computer Security Threat Monitoring and Surveillance**, James P. Anderson 1980, Fort Washington, PA: James P. Anderson Co
11. **Strategy for Handling Dead Time in Experiments**, page 2, Atlas, http://atlas.web.cern.ch/Atlas/GROUPS/DAQTRIG/TDR/V1REV1/L1TDR_Deadtm.pdf, January 2003
12. **Fundamentals of Computer Security Technology**, Edward G. Amoroso 1994, 1st Edition, Prentice Hall, ISBN 0131089293
13. **IDWG – IDMEF specifications**, http://www.silicondefense.com/idwg/, January 2003

9.2 Links

- Internet Security Systems
  http://www.iss.net
- Cisco IDS
  http://www.cisco.com/go/ids

- Enterasys IDS
  http://www.enterasys.com/products/ids
- Symantec
  http://www.symantec.com
- NFR Security
  http://www.nfr.com
- Intrusion.com
  http://www.intrusion.com
- Entercept Security Technologies
  http://www.entercept.com
- Recourse Technologies
  http://www.recourse.com
- RAID 2002 Symposium Home Page
  http://www.raid-symposium.org/raid2002/
- Snort
  http://www.snort.org

Appendix A Interview Form

This appendix covers the questions that were sent to the people interviewed. Not all the questions asked are covered here but these are the questions used to start and steer the discussion in order to get the information needed for this thesis. Keep in mind that this is a translation; the questions were originally in Swedish.

A.1 The Questions

- What does your security organisation look like and what is your part in that organisation?
- How long have you worked at your current position?
- What kind of threat is most notable for you, internal threats from inside your corporation or external threats?
- What is an intrusion detection system (IDS) to you?
- Do you use IDS today?

Have IDS today:

When

- How long have you used IDS?
- Is your IDS an own implementation or a commercial product?
- Why did you first become interested in IDS?
- How high in your organisation does the support for IDS stretch?
- How many in your organisation knows that you use IDS?
- Was there a big initial work to introduce this IDS?

Operational issues

- Do you handle the IDS by yourself or are you letting another company handle the IDS?

By yourself:

  - Why did you choose to do this by yourself?
  - Do you have staff members dedicated to this IDS?
  - If so, how many?
  - If not, why not?
  - Do you supervise the IDS day and night (24x7)?
  - How much maintenance work is required?
  - Does the IDS function well?
  - How many of the alarms you get do you estimate are false?

Others:

  - Why did you let another company handle the IDS?
  - What demands have you stated for the delivery of this service?
  - How much insight do you have in the IDS?
  - What kind of communication takes place at, e.g., a detected intrusion?

- Do you see IDS as a separate system or as an integrated part of your security systems.

- Do you produce regular reports from this IDS?
  - How are these reports used?
- What happens when you get an alarm?

No IDS today:

- Is it an active decision not to use IDS?
  - If so, why did you take this decision? Who took the decision?
  - If not, have you considered IDS?
    - If so, why have you not taken a decision?
    - If not, why not?
- Do you have any other ways of detecting intrusions?

The future

- Which problems are, according to you, the biggest of IDS today?
- What parts of IDS are most important to improve, according to you?
- Will you increase/decrease your investment in IDS? Why?
- Will your network or organisation go through any changes in the recent future?
- Do you think that IDS should be a part of your corporation's security system?
  - If not, what needs to be improved in order for you to consider that?

Other questions

- Did we forget anything?

Appendix B Abbreviations

ACL – Access Control List

API – Application Programming Interface

DMZ – DeMilitarized Zone

FIA – File Integrity Assessment

GUI – Graphical User Interface

HIDS – Host Intrusion Detection System

HIPS – Host Intrusion Prevention System

HP – Hewlett-Packard

IBM – International Business Machines

IDS – Intrusion Detection System

IP – Internet Protocol

IPS – Intrusion Prevention System

ISS – Internet Security Systems

Mbps – Megabit per second

NFR – Network Flight Recorder

NIDS – Network Intrusion Detection System

NMS – Network Management System

OPSEC – Open Platform for Secure Enterprise Connectivity

OS – Operating System

RFC – Request For Comments

SMS – Short Message Service

SNMP – Simple Network Management Protocol

SQL – Structured Query Language

TCP – Transmission Control Protocol

UDP – User Datagram Protocol

Appendix C Vendor Summary

	Cisco	Entercept	ISS	NFR	Symantec	Enterasys	Intrusion
NIDS	x		x		x	x	x
HIDS			x	x		x	x (f)
Hybrid IDS			x				
HIPS		x					x (f)
Tier	3 (a)	2	3	3	2	3 (a)	3
Usage Frequency	Cont.	Cont.	Cont.	Both	Cont.	Cont.	Cont.
Pattern Matching	x	x	x	x	x	x	x
Stateful Pattern Matching	x		x		x	x	x
Protocol Decode	x		x		x	x	x
Heuristic Analysis	x	(g)	(g)	(g)	(g)		x
Statistical Anomaly Detection					x		
Behavioural Rules		x					x
Protocol Anomaly Detection	x		x		(g)	(g)	x
Define Custom Signatures	x	x	x (b)	x	x (b)	x	
Aggregation	x		x		x	x	
Correlation			x		x	x	
Proactive Prevention		x					x
Block	x		x		x (e)	x	
TCP Reset	x		x	x	x	x	x
Redirection						(g)	
Modify User Account			x	x			
Session Recording	x		x (c)		x	x	
Trace					x		
Log	x	x	x	x	x		
Notification	x	x	x		x		x
SNMP Trap		x	x				
Spawn Process	x	x		x (d)			
Forged Responses						x	

a) The manager is accessed via a web-browser. The third-tier, or console, is therefore the browser application.
b) It is only possible to write snort rules and import these.
c) Logs all packets but without payload.
d) Aside from spawning custom processes, NFR can perform file integrity checking if TripWire is installed.
e) Manhunt will only recommend how the routers ACL could be modified to keep the attacker out. It will not perform the actual modification.
f) Combined HIDS and HIPS.
g) Reliable information concerning this could not be found.

Index

R

S

T

U

V,W