Automatic Response to Intrusion Detection

by

Pål Erik Eng and Morten Haug

Masters Thesis in
Information and Communication Technology

Agder University College
Faculty of Engineering and Science

Grimstad, June 2004

Summary

Attacks on computer systems are a growing problem. According to CERT there where 137,529 reported incidents in 2003 in contrast to 82,094 reported incidents in 2002. As the numbers of incidents grow, the work of applying countermeasures to the incidents will take more and more of the system administrator's time. To ease this job an automated Intrusion Response System (IRS) could handle some of the incident and apply the right countermeasure.

An IRS is dependent on an Intrusion Detection System (IDS), and applies responses on the incidents reported by the IDS. These responses can range from logging the incident to launching a counterattack.

In this thesis we have described IRS in general. We have also presented a new classification of IRS that classifies systems in more fine grained categories than before. Some IRSs are presented in detail. Further we have evaluated the architectures presented and refined one of them to suit a Network IDS.

The enhanced architecture includes a new decision method which can group single incidents belonging to an attack. Another feature of the improved model is the integration of a more precise IDS confidence matrix.

The framework is described in detail and we have developed a demonstrator to visualize a part of the framework.

We have proposed solutions to integrate this enhanced architecture with Telenors existing IDS, where at least one of them is feasible to implement.

Preface

This master thesis was written for Telenor Sikkerhetssystemer and Agder University College as a part of our Masters degree. The work has been carried out over a period of 20 weeks, from January to June 2004. All work has been done at Agder University College and in dialog with Telenor Sikkerhetssystemer in Arendal.

We would like to thank our two supervisors, M.Sc. Nils Ulltveit-Moe and Cand. Scient. Ole-Christoffer Granmo for guidance and support given through the project period. We would also like to thank the staff at Telenor Sikkerhetssystemer for their support and feedback.

Grimstad, June 2004

_____ _____

Pål Erik Eng Morten Haug

Table of Contents

List of Figures

1 Introduction

1.1 Background

As time passes, society gets more and more dependent of computers [18] [24]. Therefore, the degree of seriousness of a computer attack increases proportionally. According to CERT[1] statistics [15], it has been an exponential growth of reported attack incidents since the institute started its statistics in 1988. There is no reason to believe that this trend will change very soon. CERT also publishes annual and quarterly reports and summaries. These reports confirm that the attacks are more frequent and more sophisticated and that worms are the big hit of 2003 [16].

From CERTs statistics, another trend which is more subtle is that with the increasing number of incidents reported, we can conclude that more attacks are in fact detected. But there is still a lot to be desired from Intrusion Detection Systems (IDS) in general and Intrusion Response Systems (IRS) in particular.

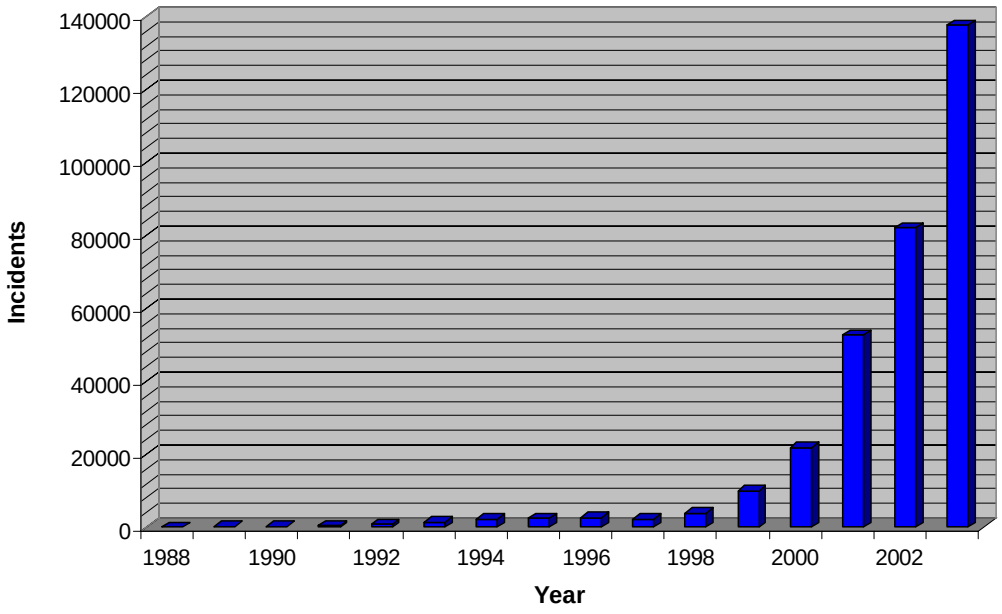

Figure 1: Annual reported incidents at CERT 1988-2003

Intrusion Detection Systems have a large following and have been the subject of much research the last decades. L. Mé and C.Michel published in 2001 a bibliography [21] containing over 600 references to intrusion detection papers. The story is different however, with intrusion response research.

The need for prompt responses is explored by Cohen in [17], where the results indicate that this is of great importance. It states that if the delay[2] from the detection to the response is 10 hours, then a skilled attacker will have 80% chance of succeeding. If the

[1] CERT is a major reporting centre for Internet security problems and has been a source of security related information since 1988.
[2] This delay is also known and referred to as window of opportunity.

attacker is given 20 hours he will succeed 95% of the time. Another interesting conclusion is that if the attacker gets thirty hours or more, the skill of the system administrator does not have any bearing on the outcome. Last, but not least, a valuable finding for this thesis was that if the response was immediate, then the attack would almost never succeed.

1.2 Intrusion Detection Systems

As early as 1980 Anderson [8] introduced the possibility to use computer systems to establish "normal" behaviour. In his paper he described several system intruders; the external penetration is defined as a person who successfully gains access to the system but is not an authorized user on the system; and defined internal penetration as when a legitimate user of the system gains access to resources or services he is not authorized to access. In the latter penetration class, Anderson identified three classes of users; the masquerader, the legitimate user and the clandestine user. In the same paper Anderson proposed how these intruders should be detected.

While Anderson's paper introduces intrusion detection on a single host, Denning [9] presented in 1987 a model for an intrusion detection system for use in networked environments, which has been a cornerstone for IDS development.

In more recent time the evolution of Intrusion Detection Systems has forced on. The trend in the later years is that the research is focused at developing specific areas. Many research projects are based on the work done on MIT Lincoln Laboratories in the DARPA Intrusion Detection Evaluation project [19].

Thomas Toth presented in May 2003 a dissertation named *Improving Intrusion Detection Systems* [12] on signature based systems. Here he presented contributions to the solution of three important problems of current intrusion systems. First he proposes a way to efficiently handle incoming events at a high speed and comparing them with signatures, by applying clustering of the incidents. Then he present a solution to the problem that signature based systems often cannot detect a slightly modified signature. This is solved by introducing abstract signatures. Toth also presented a method which allows administrators of an IRS to define the objective of the network and the internal dependencies.

1.3 Intrusion Response Systems

Even though responding to incidents is an important aspect of security, it has not been given the same attention and research activity as detecting them. This applies in particular to automated response actions.

In 1989 one of the first methodologies for incident response was created at *Invitational Workshop on Incident Response* at the Software Engineering Institute in Pittsburgh, Pennsylvania. It is later reused by one of the original workshop participants, Schultz, in his book [10] from 1989 on incident response, so not much has changed over that period. According to his book, this approach has been the most time-honoured methodology.

The methodology is a six-stage plan to deal with incidents; preparation, detection, containment, eradication, recovery, and follow-up.

CERT has also published a report [11] which is supporting the general idea, but also elaborates some of the work done by Schultz and the workshop.

Automated response actions are today implemented in some IDSs. Toth [12] presents 14 IDSs which have some sort of automated response (as of May 2003), but all these have very limited functionality.

Some good work has been done in the field by Carver [1] with his AAIRS architecture, which provides a framework for other implementations of intrusion response systems.

There has also been some research in the area of modelling the costs of the response actions, namely by Toth [12] and Lee et al [14].

1.4 Thesis definition

We wanted to take a closer look at the research done in the field of Intrusion Response Systems and to dissect this part of computer security. In collaboration with Telenor Sikkerhetssystemer[1] we also found out that we should combine the existing Intrusion Detection System currently deployed by Telenor with an Intrusion Response System. Therefore, the final thesis definition was defined as follows:

"Intrusion Detection Systems have been around for some time. An extension to the actual detection would be to respond automatically to an intrusion. In order to remedy this, Intrusion Response Systems have come to be an important research issue. The thesis should give an introduction to Intrusion Response Systems (IRS) and make a brief summary of the technologies available today. Based on the summary, choose an IRS technology to extend Telenor Sikkerhetssystemers IDS architecture, if this is feasible. If not, suggest an architecture that may include both IDS and IRS.

Preferably and if time allows it, a demonstrator of the architecture should be implemented, tested and verified against earlier assumptions."

There has been no change in the thesis title, so it has stayed the same during the whole thesis period:

"Automatic Response to Intrusion Detection"

1.5 Work description

Currently, the research efforts in the field of IRS are scarce and scattered in minor projects. One of our main tasks is too look into this field in detail to see what have been done and make up status on the latest advances.

[1] For the English reader: Sikkerhetssystemer is Norwegian and means Securitysystems.

Since the thesis definition was given in collaboration with Telenor, it has been in our common interest to take a closer look at their existing system for intrusion detection to use as a basis for the intrusion detection part of our proposed system.

With the status report on the current advances, we will choose technologies which are considered to be the best and which complements Telenors existing system.

Based on these considerations we will propose a combined framework for a complete Intrusion Detection & Response System (IDR), which can be used in general and in particular by Telenor Sikkerhetssystemer.

1.6 Report outline

In the following chapters we will take a closer look at the technologies mentioned in the above sections. We will give a further introduction to Intrusion Detection Systems in chapter 2, before we start the intrusion response section.

An introduction to Intrusion Response Systems is given in chapter 3, before we take a look at research done in this field in chapter 4. Chapter 4 also includes a brief summation of what is already available on the market today. In chapter 5, we do an evaluation of three eligible Intrusion Response Systems.

A response taxonomy is presented in chapter 6, and our conceptual IRS is introduced in chapter 7. A prototype of the proposed IRS is described in chapter 8. In chapter 9, we propose different integration solutions with Telenor IDS.

In the end of this thesis, we have discussions in chapter 10 and the final conclusions in chapter 11.

2 Intrusion Detection primer

2.1 Introduction

"Intrusion Detection System is defined as any hardware, software, or combination of thereof that monitors a system or network of systems for malicious activity." [7]

A very good and often used analogy is a burglar alarm. You have sensors monitoring activity, and an alarm is generated whenever misuse is detected.

An IDS is often, by novices, considered to be a complete security structure. This myth should be put down, as the analogy continues to hold true. Without proper physical barriers (brick wall vs. firewall) an intruder has no problem getting what they want. He could not care less if an alarm was triggered.

IDSs are passive by nature, as they just monitor traffic, so they have to be supplemented with response actions. The analogy between police/security firms and response action entities is a natural parallel.

Since the performance of the IRS is directly dependent on the quality of the IDS it is necessary in this chapter to take a closer look at these systems.

2.2 IDS flavours

2.2.1 Host-based IDS

Host-based IDS (HIDS) resides on one machine and monitors that specific machine for intrusion attempts.

Typical measurements a HIDS will monitor are logs, error messages, service and application rights, and resource usage on a host.

In comparison with Network-based IDSs (NIDS), HIDS are more accurate at detecting genuine intrusions because of their relatively lower rate of false-positives [7]. Another advantage HIDS has over NIDS is the extended knowledge of the host it monitors, and it can detect attacks which can seem like normal traffic to a NIDS.

The disadvantages are significant however, as they reside on the monitored host. This limits the view the IDS has of the network topology, and it cannot detect attacks on other hosts. The result is that every host has to have a HIDS monitoring them, which in most cases would not be feasible and in some cases impossible.

2.2.2 Network-based IDS

NIDS are put into the network infrastructure at strategic locations, where they monitor passing network traffic. This type of IDS is much more cost effective than a HIDS as you can monitor several hosts with one, which also gives a better overall picture of the threat situation.

Another advantage is that the NIDS are separate entities in the network and can be tightly secured. Therefore, they are less prone to be compromised.

A downside to NIDS is the need for very efficient packet scanning, so that no threats are missed going through the network. With the increasing bandwidth and internet usage on everyday basis, the amount of network traffic is making it tough for the IDS to keep up.

The other main disadvantage is the numerous methods hackers have discovered to hide malicious network traffic; these methods are also referred to as IDS evasion techniques.

2.3 Detection methods

2.3.1 Signature detection

A signature represents a known intrusion. These signatures are then compared to current system activity and raise an alert if it is a match.

Signatures are created to suit known intrusion characteristics and with the most common NIDS today, a large database with the common signatures are bundled. New signatures are also made as new attacks are discovered to these NIDS.

Intrusion detection based on signatures is currently the most accurate method to detect known attacks. Every time a signature is matched, an alert is raised and countermeasures can be taken.

An obvious drawback is that this only detects known attacks, so new attacks will pass right through without raising an alert. A change of only one bit of the attacking package could be enough to defeat the IDS.

Another drawback is that the IDS has no idea of what the intention of the network activity are. In many cases, normal traffic might look suspicious and will then raise alerts. This is also called a false positive.

According to Koziol [7], IDSs which utilize signature detection are the most prominent and reliable on the market today.

2.3.2 Anomaly detection

In anomaly detection, you record and establish a normal pattern of usage over time, and when the pattern is different from this norm an alert is raised.

The greatest advantage of this method is that it does not rely on having previous knowledge of an attack, it just responds to deviations in normal use of the system.

A downside to anomaly detection is that it has to have a training period in which it will learn what normal traffic should look like. If you do not have full control over the traffic in these periods, regular attacks would be characterized as normal traffic and will not raise an alert later on when the system is deployed.

The same problem applies to safe and normal traffic that rarely happen, which will trigger an alert because it was not established as normal in the training period.

The latter disadvantage mentioned is the main reason why signature detection is the preferred methodology.

2.4 Summary

Intrusion Detection Systems are an important part of a computer security structure. It is important to remember that such systems only report when malicious content arrives and is seldom capable of countering it.

There are several flavours of IDSs, with the two main categories location and detection method. These categories are equal, meaning an IDS has a location and a detection method.

In the location category, we find the Network IDS which is placed in a network to sniff traffic and detect malicious activity in network packets. Here we also find the Host IDS which is placed on separate hosts and can detect abnormal use and monitor other system specific attributes.

In the other main category, there are two methods of detecting an attack. Signature based detection utilizes a database of known attack signatures which constitutes the rules. Whenever a network packet is matched for any of these signatures, an alert is raised. Anomaly detection takes a different approach; it defines what normal traffic look like and then raises an alert whenever abnormal traffic appears.

In the next chapter we will start the main events of this thesis; we are entering the intrusion response domain.

3 Intrusion Response primer

3.1 Introduction

If you have an IDS without any form for output, then the system would be useless. You need to have some sort of output from the IDS in order to analyze and respond to the intrusions. To do this, different solutions has been applied, and Carver [1] provides a categorization for Intrusion Response Systems; notification, manual response and automatic response systems.

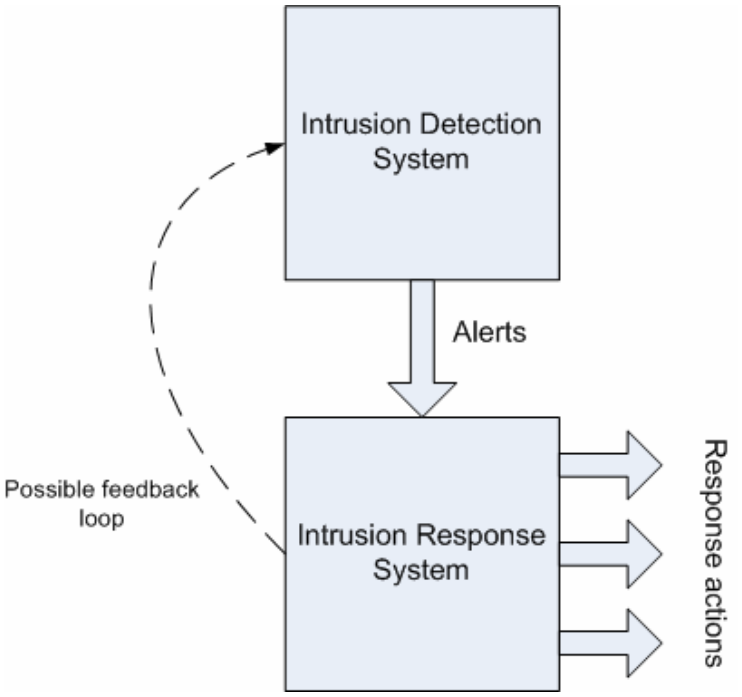

Figure 2: General IRS functionality

In addition to Carvers categorization, we have made an addition to his work, by specifying one of his main categories even more.

In this chapter we will first comment Intrusion Prevention Systems, and then look at how incidents traditionally have been handled manually. After that, we present Carvers categorization along with our own proposed categorization. Last in this chapter, we give an introduction to attack- and response-taxonomies.

3.2 Intrusion Response or Intrusion Prevention?

The latest buzzword in the intrusion detection field, is definitively Intrusion Prevention Systems (IPS).

The terms IRS and IPS are often misleadingly used as if they were the same thing, but this is not the case.

The connection between the two is that IPSs are a subset of Intrusion Detection and Response Systems. Intrusion Prevention Systems are souped-up IDSs, with the ability to

stop packets on the fly whenever an attack is discovered. Other mechanisms include setting temporary rules in the firewall.

3.3 Incident handling

In section 1.3 we only scratched the surface on how a response could be handled. In [23], they present a method of handling events following intrusion detection in an orderly fashion:

1. **Initial analysis** is done immediately after detection. It includes crude estimation of the inflicted damage, the source of the intrusion and if possible, the initial entry method. It is meant as a quick damage assessment step.

2. **Containment** is the process of stalling and delaying the intrusion attempt in order to limit the damage.

3. **Damage assessment** is a time intensive task, and the time to do this is not available before this step. This is needed to make a more detailed analysis of the incident.

4. **Restoration of operations** is done after the thorough analysis. The system(s) involved will be rolled back to the last "known good" state, i.e. reinstalling the operating system or load a backup.

5. **Process or mechanism correction** should ensure that the vulnerability exploited by the attackers disappears. This may include patching up services and upgrading software. It may also include updating company policies for security measurements.

6. **Recovery and summation** is the last task employed after an attack. A final and thorough report on the incident should be generated. This step may also include legal actions against offending parties.

These tasks are typically done manually, and in our case, they are distributed between the security expert at Telenor and the company administrator. As we can see, it is time consuming to respond to an incident, so one man can not handle many incidents in a good way. Some sort of automation would ease the burden considerably.

It is important to remember however, that not all tasks can be done by an automatic system, but most of the initial tasks should be possible and feasible with todays technology.

3.4 Categorization

In current literature [1] [12], a main categorization is made based on principal approaches for intrusion response. The three approaches are notification, manual response, and automatic response systems.

We found however, the current categorization to be too coarse, so we divided the main category of automatic response systems into three subcategories. These new

subcategories of automatic response systems are association based, expert based (stateless), and adaptive based (stateful) systems. Figure 3 shows how the complete categorization.

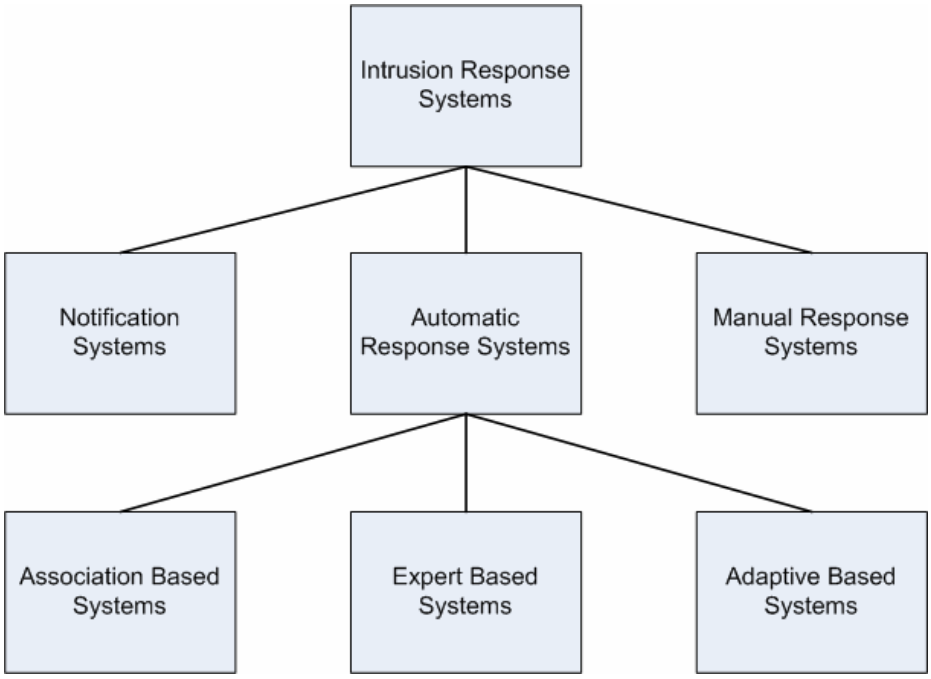

Figure 3: Intrusion Response System categorization

In the following sections a more detailed description of the different systems are presented.

3.4.1 Notification Systems

These systems just generate reports and alerts. This is the most common category, where most "classical" IDSs fit, like Snort without any special output-plugin. With predefined intervals, the IDS reports incidents and raise alerts. They can take the form of a web interface, an email message, or another paging service, i.e. SMS. The response is then the responsibility of the administrator receiving the alert.

The great problem of this approach is the time delay between the actual intrusion and the actual response to encounter this intrusion. Given a standard work-schedule for a system administrator, an attack going on through the weekend will be detected and dealt with accordingly, in a worst case scenario, on Monday morning.

A second problem is the time used by the administrator in the task of evaluating and eventually responding to the alerts. Administrators today often have a wide range of tasks in his/hers company, and time used on these measures is most certainly taking time from other duties. This is often the situation in small and medium sized companies, having only one employee in the computer department.

Special knowledge is also needed to analyze and respond correctly to the various threats; this is a demanding task for a one-man show.

3.4.2 Manual Response Systems

Manual response systems have evolved from the traditional reporting structure of notification systems. These systems should recognize the type of alert being raised, and provide preconfigured response alternatives to the administrator. The responsibility is still placed at the human administrator part of the system, but is more user-friendly and provides responses speedier than systems with notification only.

As with notification systems, this approach has the same time delay between detection and handling of the attack. These systems however, speed up the process of selecting and performing the appropriate response actions.

3.4.3 Automatic Response Systems

These systems replace the administrator with a decision making device. This means that if an alert is raised by the IDS, a response is issued immediately.

The time delay is now close to zero, so this is not a problem anymore. Another problem arises though, as computer decision devices as of now are not as intelligent as a human administrator, and therefore are more suspect to response inadequately or worse; respond incorrectly.

Based on the response system complexity, it is useful to divide this class into three subclasses; association based, expert based, and adaptive based systems.

Association based systems

These simple decision devices do not really make any decisions; they simply associate a specific alert to a specific response. Whenever a specified attack occurs, a response will be triggered.

This simple approach is very vulnerable to attackers, since they can easily adapt to a static response strategy and even take advantage of it.

Expert based systems (stateless)

These systems have actual decision making devices. The ways decisions are made are different from system to system. The devices in the complex range base their response action decisions on one or more metrics, i.e. the intrusion severity and threshold metrics in EMERALD [2] [3] or the suspicion level and DC&A taxonomy in CSM [4] [5].

However, these systems do not learn anything from attack to attack, so they cannot increase their artificial intelligence level during their lifetime.

Adaptive based systems (stateful)

Adaptive based systems have a notion of learning. This means the systems have feedback loops to evaluate the decisions it makes and strengthen or weaken its confidence that it made the right decision.

There are just a few proposed adaptive-based systems. Carvers Adaptive Agent-based Intrusion Response System [1] (AAIRS) is one such proposed system. It applies

learning with the update of a confidence metric (how confident the IRS is that the IDS raised a correct alert). This learning is however, not automatic, but updated by humans.

It has also short term learning, with its capability to adapt its response plan.

3.5 Taxonomy

To be able to launch a response to an attack, it is necessary to determine what kind of attack we are dealing with. The classification of attacks is done by an attack-taxonomy. To determine witch response that is most feasible a response-taxonomy is needed. In this section we will describe the Lindqvist attack-taxonomy and a response-taxonomy presented by Carver [1].

3.5.1 The Lindqvist Taxonomy

The Lindqvist Taxonomy [13] is a classification made from the user's point of view and uses intrusion techniques and intrusion results to classify the attack. The taxonomy is based on a scheme proposed by Neuman and Parker in 1989. It divides the type of attacks in three classes; *exposure*, *denial of service* and *erroneous output*, which is taken from the traditional computer security theory; confidentially, integrity and availability (CIA).

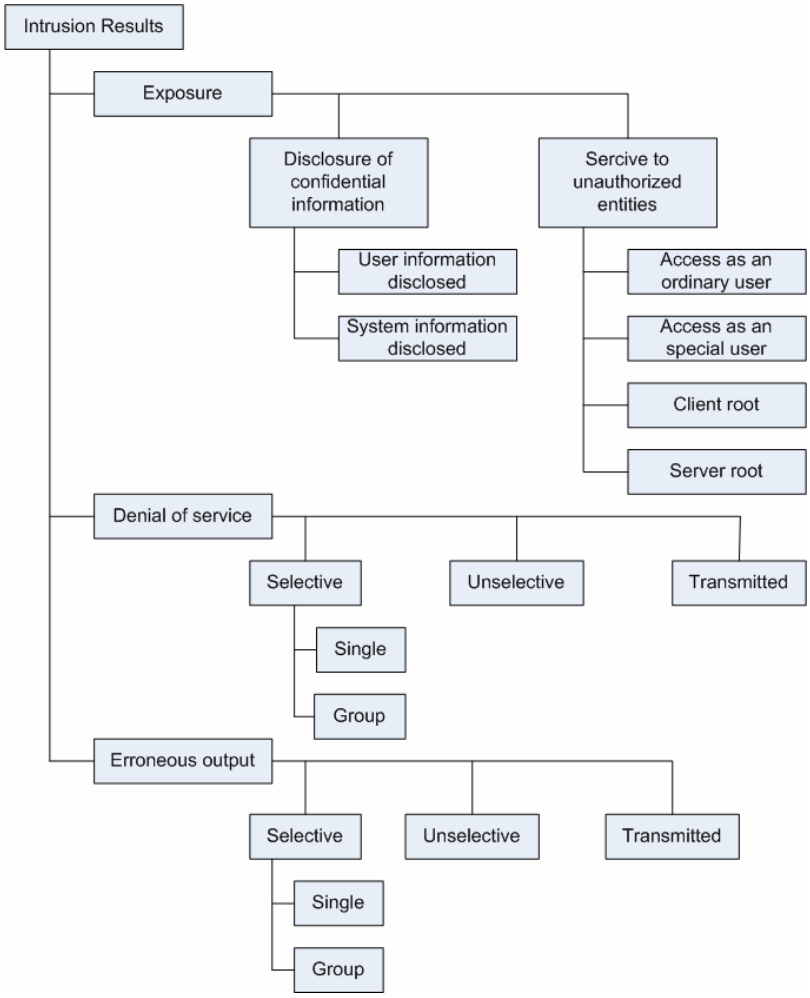

Figure 4: The Lindqvist Taxonomy

Exposure

Exposure attacks are directed against system confidentially. More precise it means attacks to get unauthorized access to confidential information.

The *exposure* class is divided into two subclasses; *disclosure of confidential information* and *service to unauthorized entities*.

Disclosure of confidential information is further divided into *user information disclosed* and *system information disclosed*. An example of disclosure of confidential information is reading backup tapes and password files.

Service to unauthorized entities includes sub-classes to reflect privileges associated with the service delivered. Examples of service to unauthorized entities are automated password- guessing and manipulating the boot process. The sub-classes are:

- *Access as an ordinary user* is a legal user that gains access to another user account, or an outsider who gains access to any user account on the system.
- *Access to a special system account* is when a user gets access to an account with higher privileges than an ordinary user account.
- *Access as client root.* The client root has no special privileges on the server.
- *Access as server root.* The server root has special privileges on the server.

Denial of Service

Denial of Service are attacks directed against system availability and are again split up in three subclasses; *selective* (includes the sub-classes: user and group), *unselective*, and *transmitted.* Intrusions that affect services provided from other systems, belongs in the transmitted class. An example of a denial of service attack is causing a system to crash by forcing a remote copy to an audio device.

Erroneous output

Erroneous output is defined as attacks directed against the system integrity. I.e. output on user terminal, network, files on hard drive and memory. Like the *Denial of Service* class, it is divided into *selective*, *unselective*, and *transmitted* sub-classes. Examples of this class could be the attacks called Spoofing Xterm and Faking e-mail.

3.5.2 Carvers response-taxonomy

In his AAIRS dissertation [1] Carver presented his own response-taxonomy. To perform an automated response a categorization of possible offensive or defensive responses must be done. Carver presented a taxonomy consisting of six dimensions; *response timing*, *type of attack*, *type of attacker*, *strength of suspicion*, *implication of the attack*, and *environmental constraints*.

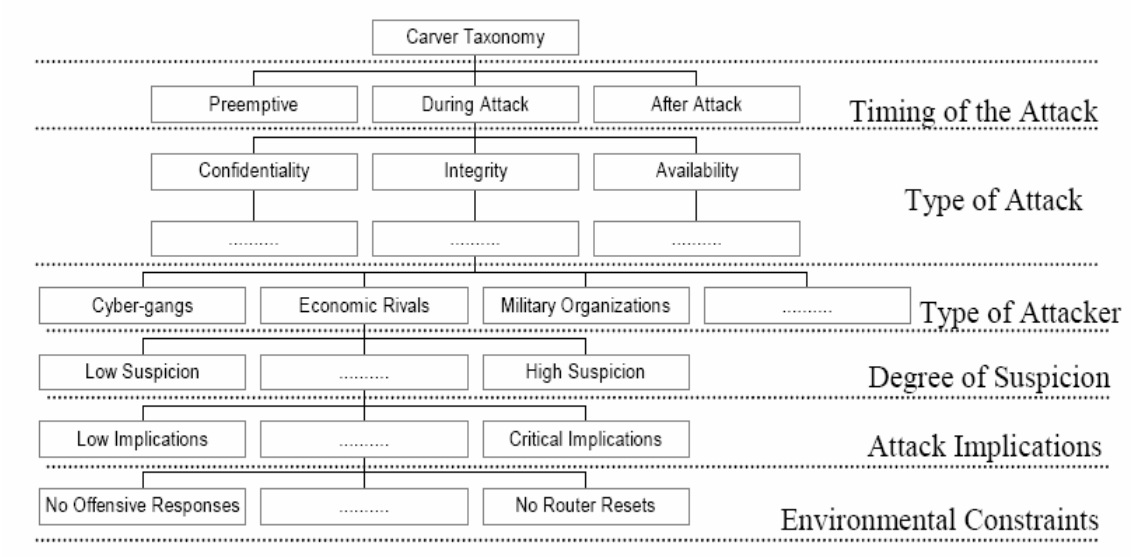

Figure 5: Carvers Response Taxonomy

Timing of the attack is defined as *pre-emptive*, *during an attack* or *after an attack*. For determining the *type of attack* the Lindquist Taxonomy is used. The *type of attacker* is essential to apply the right response. Type of attacker can be *novice* or *expert* and *manual* or *automated*. The *strength of suspicion* is used to eliminate the possibility of launching a hard response on a false positive. The possible responses is limited if the strength of suspicion is low. With the *implication of the attack* dimension the possibility of different systems having different degrees of importance. *Environmental constraints* are the last dimension. Here legal, ethical, institutional and resource constraints are considered.

3.6 Summary

In this chapter we have given an introduction to Intrusion Response Systems. We have looked at what should be done in the event of an incoming incident, these steps are; initial analysis, containment, damage assessment, restoration of operations, process or mechanism correction, and recovery and summation.

There are three main categories of response system approaches; notification, manual response, and automatic response systems. Furthermore, we have made a contribution to this de facto categorization, and split up the automatic response systems into more fine grained categories. These subcategories are the association based, the expert based, and the adaptive based systems.

In this chapter we have also described a way to classify attacks, the Lindqvist attack-taxonomy, and way to classify responses, the Carver response-taxonomy.

In the next chapter we will do a thorough literature review, to see the IRS research situation today.

4 IRS literature review

4.1 Introduction

In order to evaluate the most current and applicable technologies, these must be found in the research papers, articles, and white papers written.

We have divided this chapter in three main parts; the first section is a review of research done. The second and third section will comment some of the free and commercial products available.

4.2 Research related to IRS

There have been some papers listing current IRSs. The most comprehensive and thorough paper, is Carvers survey from 2000 [25], where he comments 56 systems.

From his findings, there were no dedicated solutions for intrusion response. There were however, some responses implemented in a variety of IDSs. Most of the IDSs where notification and manual response systems, which are not preferable solutions. There were however, some automatic response systems as well, but these were rather immature.

In 2001, Carvers Ph.D. dissertation on the subject [1], proposes a system for intrusion response, namely AAIRS. This is the first response system implementing a notion of learning that we know of.

At the same time, at Lincoln Laboratory, MIT, Lewandowski et al. introduces a paper on *SARA: Survivable Autonomic Response Architecture* [26]. This paper introduces an architecture with coordinated autonomic responses.

In this paper, an analogy to nature's autonomic responses is drawn. There have also been other papers which uses analogies to nature as a solution to computer security problems. A very interesting paper on immunology [27] was presented by Mark Burgess already back in 1998. Here, he draws the analogy between biological and social systems, and computer systems.

This approach is also used by Fenet and Hassas in their paper [28] from 2001. Here, they use a distributed model of agents to detect and respond to intrusions in an analogy to insects.

Lee et al. [30] provides in 2000, a good introduction to modelling costs in intrusion detection and responses. These cost factors can be qualified according to a defined attack taxonomy and site-specific security policies and priorities. This is an important factor when dealing with responses which can have negative effect in its own network.

An adaptive IRS is presented by Tanachaiwiwat, Hwang, and Chen in 2002 [29]. It introduces collective risk assessment of multiple attacks on a network. They explicitly target to circumvent human interaction with the system, which should be a common

goal for all intrusion response research. Their main contribution is the alarm matrix, which holds information on how good each IDS is on different types of attacks. This extends Carvers system where he only has one metric for the whole IDS. This work also builds on Lee et al. cost modelling.

In 2002, Toth and Kruegel [31] use cost as well as dependency models to find out impacts of automated intrusion response mechanisms. Then, when the IRS is presented with several response actions, it can make its decision based upon these dynamic dependencies.

The most recent work is done by Wu et al. with their ADEPTS [32] system, which is closely related to Carvers AAIRS. Each intrusion is considered to have an ultimate goal, and several minor goals along the way. They use a directed acyclic graph to represent these goals.

4.3 Freeware applications

There has been developed some application for the Linux platform, which are free to use.

Hogwash [36] is an IPS solution with a modified version of Snort as basis for the detection part, so it has a signature based engine. To stop attacks it has the ability to drop or modify specific packets based on a signature match. Another advantage Hogwash has is that it resides right on top of the network driver with no need for an IP stack to work. This means it can stop attacks that can not be blocked by a traditional firewall and to protect unpatchable systems.

Snort_inline [37] is also an IPS based on a modified version of Snort. It accepts packets from iptables, which is a firewall for linux. It then uses new rule types to tell iptables if the packet should be dropped or allowed to pass based on the snort rule set.

4.4 Commercial solutions

As mentioned in section 3.2, Intrusion Prevention Systems are getting a foothold in the security market. Here we will give a brief overview of some of the solutions available today.

UnityOne [33] is an IPS solution from TippingPoint Technologies. It works as a filter, which will detect and remove malicious traffic and attacks. These include trojans, worms, viruses, and DoS-attacks. This is a complete appliance to be installed into your network.

Border Guard [34] is Latis Networks' IPS solution for Linux. It has the same functionality as UnityOne; it can drop packets instantaneously to ensure no attack reaches the network it is monitoring.

Network Associates Technology is a well known security company, and they have an IPS available. Their product line of IPSs is called IntruShield [35], and can cope with gigabit-speed networks. The IntruShield architecture integrates signature, anomaly and DoS analysis techniques. To deal with attacks, it has the capability to drop offending

packets, change firewall rules, and generate notifications and alerts. All this happen in real-time.

4.5 Summary

We have in this section presented what has been done in the intrusion response research area. The earliest forms of automatic responses were simple association based and expert based systems. This makes the basis for the first adaptive based systems, with AAIRS and SARA as early architectures. These have later been refined, but their principles are still important for this research.

Snort has been used as the basis for both of the mentioned free applications for intrusion prevention. This means both of them rely on Snorts signature based detection engine.

The commercial IPS systems all have the same attributes; drop attacking packets, insert firewall rules, and notify administrators. There are just minor differences in approaches.

From the documentation available on all the IPSs mentioned, we can categorize them all as automatic response systems. Further categorization however, puts them all in the association based system category, described in section 3.4.3.

The next chapter will give a more detailed description and evaluation of eligible technologies.

5 Evaluation of existing Intrusion Response Systems and Architectures

5.1 Introduction

Telenors architecture has the IDS portion of a complete Intrusion Detection and Response System in place. So our main work is to find a complementing response system.

It should be a stand-alone system, which can have either a specific Snort interface or a general IDS interface for input. Another main criterion is how the alert is handled when it enters the system; there should be as little as possible of human interaction with it.

As a third main criterion, environmental and historical learning should also be applicable.

In the next sections we will do an evaluation of the most current and appropriate systems.

5.2 Snorts Flexresp2 module

Jeff Nathan has made a native response module [22] for Snort. This module will allow users to configure rules that will attempt to actively terminate connection attempts.

5.2.1 Functionality

In appendix A, we explain how rules for Snort are written. To utilize FlexResp2 active responses, rules are created with inclusion of the resp keyword. The following modifiers are allowed for this keyword:

- **reset:**
  send TCP reset packets to the receiving system (a minimum of three packets using a shifting ACK number)

- **reset_attackresp:**
  send TCP reset packets to the sending system – for use only with attack-response rules (a minimum of three packets using a shifting ACK number)

- **icmp_net:**
  send an ICMP network unreachable packet to the sender

- **icmp_host:**
  send an ICMP host unreachable packet to the sender

- **icmp_port:**
  send an ICMP port unreachable packet to the sender

- **icmp_all:**
  send all three ICMP unreachable packet types to the sender

In addition, one or several packets will be generated in an attempt to actively terminate the connection.

An example of a rule with the resp keyword (in bold) is as follows:

```
alert tcp $EXTERNAL_NET any -> $SMTP_SERVERS 25
(msg: ”SMTP HELO overflow attempt”; flow: to_server,
established; content: ”HELO”; offset: 0; depth: 5; content:
!”|0a|”; within: 500; reference: bugtraq, 895; reference: cve,
CVE-2000-0042; reference: nessus, 10324; reference: bugtraq,
7726; reference: nessus, 11674; classtype: attempted-admin; sid:
1549; rev: 11; resp: reset;)
```

Whenever this rule is matched by an incoming packet, the Snort detection engine will raise an alert as well as making an automated active response to the intrusion attempt.

5.2.2 Evaluation

This particular response mechanism is highly static and would be categorized as an association based system described in section 3.4.3.

The system is also resource consuming, in that the machine running Snort, will do the actual transmission of counter-packets. This will prohibit the continuous sniffing and capturing of incoming packets, meaning there will be small timeframes where the system will be vulnerable for new attacks [22].

On a good note, this system is easily implemented with Snort, and could be deployed instantly. The only thing, which would only be done once, is to reconfigure the Snort rules which should react in this manner.

The human interaction factor is mediocre, since Snort regularly comes with updated rule sets, and sometimes completely new rulesets, so a rule maintenance job where a large number of rules needed patching would cause considerable work to maintain, unless it got automated.

5.3 SARA: Survivable Autonomic Response Architecture

In a paper [26] from 2001, S. M. Lewandowski et al presents an architecture for coordinated autonomic response actions.

They focused on the DARPA/ISO Autonomic Information Assurance (AIA) program, which had led to two hypotheses; fast responses are necessary to counter advanced cyber-adversaries and that coordinated responses are more effective than local reactive responses.

5.3.1 Functionality

SARAs inner and outer loop functionality, which is shown in Figure 6, is designed to test both hypotheses.

In many cases, local responses to attacks are useful. In most cases however, a global view of the network is needed to maintain a better state for all nodes involved. An example would be a DNS server shutting down its service due to consequent attacks. This would penalize the other network nodes so harsh most nodes will be rendered useless.

When choosing countermeasures to be deployed, SARA takes into account what resources which will be affected and the resulting impact on the mission.

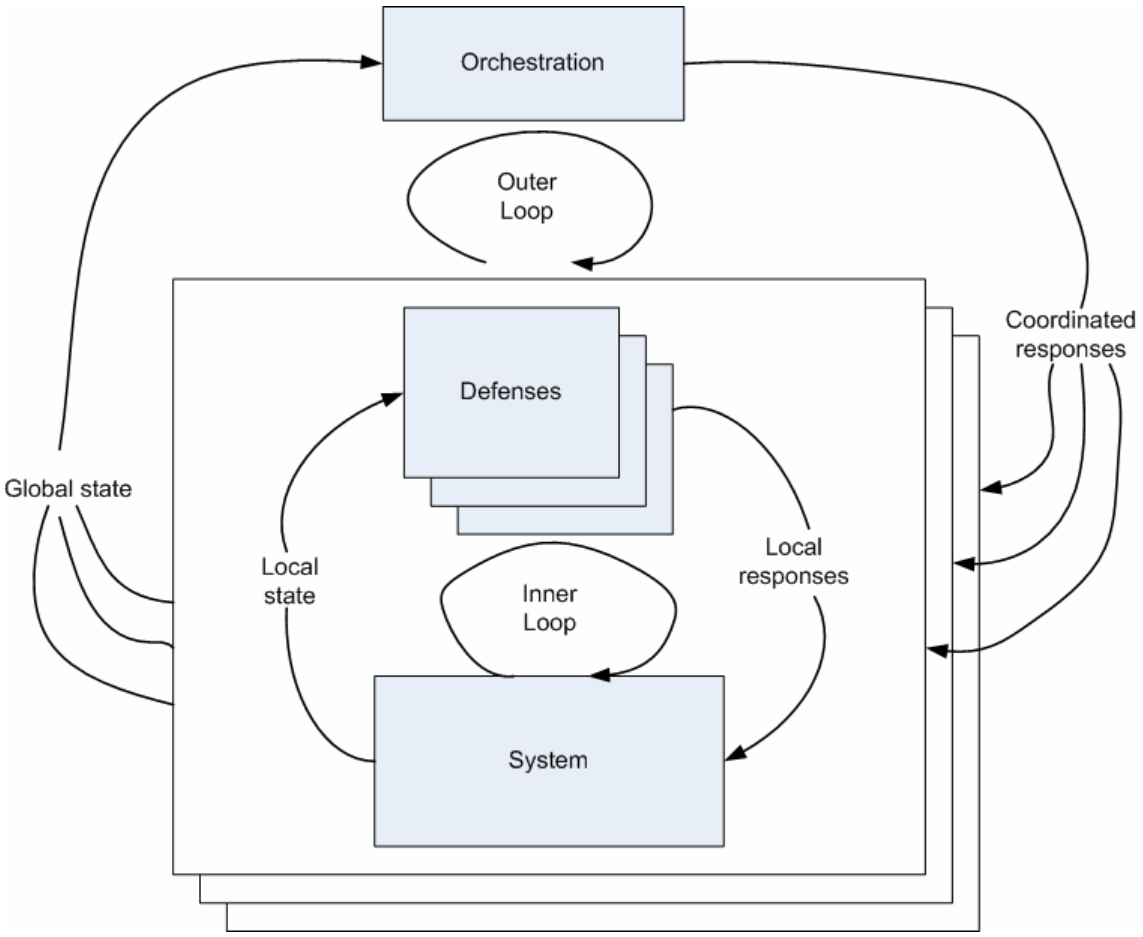

Figure 6: SARA inner and outer loops

The SARA architecture consists of a set of components that provide important capabilities to a system and a set of interfaces to those components. The main components and interfaces are shown in Figure 7.

The SDAR[1] components are the defensive parts of the system and are classified as follows:

- **Sensors:**
  These components gather data about the system. Sensor data consists of reports of events and status. These data are objective.

[1] SDAR is short for the four components; sensor, detector, arbitrator, and responders.

- **Detectors:**
  The detectors analyses the data from the sensors and indicate if it is an attack or not.

- **Arbitrators:**
  The decision making about what should be the response is done here. Its decision may be based on the sensor data, the detector alarms, and other system states available.

- **Responders:**
  The actual response is executed by the responders. A toolkit can be provided where several responses are available.

These components can be legacy components as well as specially crafted SARA components.

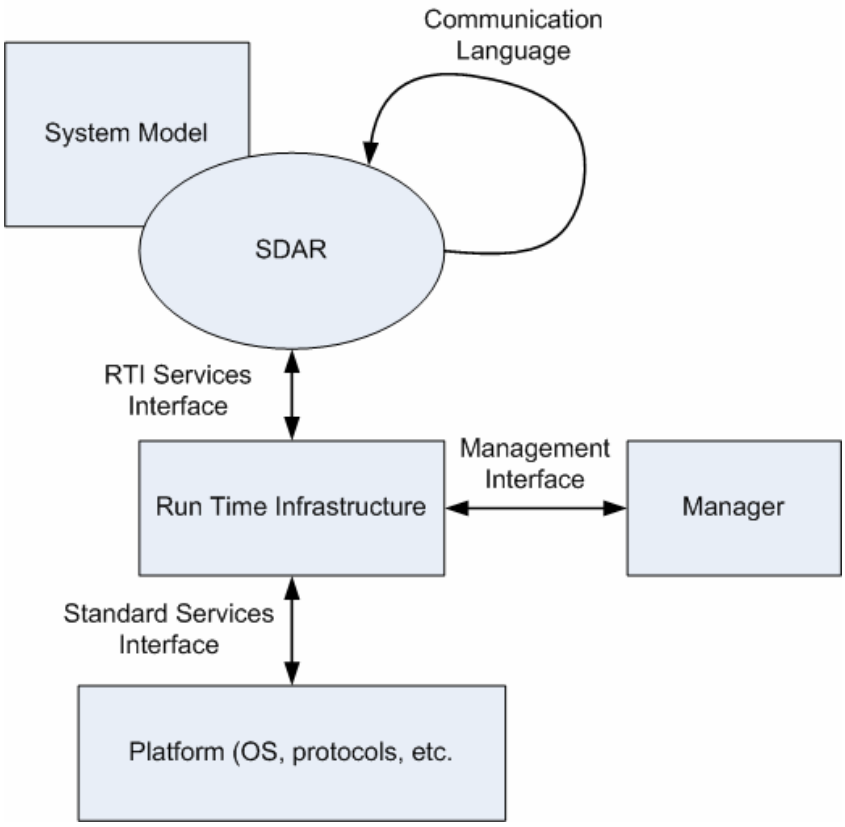

Figure 7: SARA components and interfaces

The Run-Time Infrastructure (RTI) provides communication and coordination services to SDAR components. It works as a middleware layer between these and the system platform. As a middleware, it hides platform specific quirks and will work when deployed in a heterogeneous environment. All communication between SDAR-components should go through RTI, but legacy software is allowed to use private communication channels.

To facilitate effective decision-making, the System Model is used to capture the context of the defended system. It also contains a description of the system characteristics. Typical characteristics in the model might include the network topology and applications running on nodes in the system.

Managers provide monitor and control functions for human surveillance and configuring.

5.3.2 Evaluation

The SARA architecture provides us with a conceptual approach to how a response system could be developed. It forms a basis for others to develop working components.

It does not give any guidelines on how the SDAR components should behave, except for the communication via RTI. A typical SDAR combined sensor and detection component would be a NIDS, and a combined arbitrator and responder might be an IRS.

As such, it cannot be labelled an IRS. Therefore, a categorization from section 3.4 cannot be applied.

This architecture however, does give a good infrastructure basis for a system of different components. It also gives good guidelines on how the information flow should go between entities in the response system.

5.4 AAIRS: Adaptive Agent-based Intrusion Response System

In his desertion from 2001 Curtis A. Carver Jr. presented a framework on an adaptive response system [1]. This framework is designed for host-based intrusion systems. The design includes intrusion response taxonomy and associated methodology, where the taxonomy provides a theoretical classification of responses and the methodology describes the conceptual model of the system.

5.4.1 Functionality

The model is divided in modules, as seen on Figure 8 below, where the different modules handle a specific task. The functions are described briefly underneath, for a detailed description see [1].

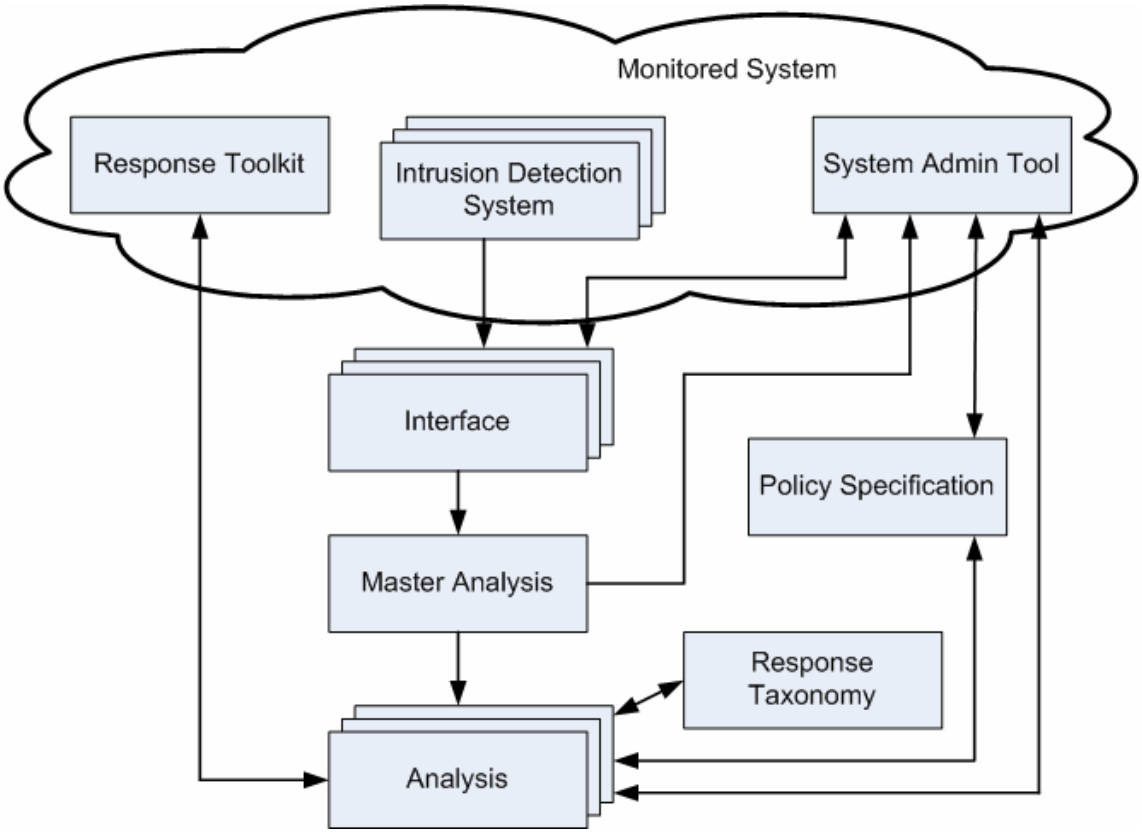

Figure 8: AAIRS components

The *interface* component task is to translate IDS specific messages into messages on a generic form, called *incident report*. Also included in the *interface* component is an IDS confidence metric. This metric is the ratio of false positives to actual reports for the different IDSs. The metric is updated manually through the *system administrator tool*. The metric is passed on to the *Master Analysis* module.

The *Master Analysis* receives an incident report and determines if it is a new attack or a continuation of an existing attack. To determine if an incident is a part of an ongoing attack three metrics is taken into consideration: time, session identifier, and attack type. If an incident report is found to belong to an already monitored attack, it is forwarded to the specific *analysis agent*. If an incident is not a part of an already monitored attack a new *analysis agent* is created.

The monitoring of each attack is handled by the analysis components, who determine a plan for responding to an attack. In the response plan the system administrator has specified what the response goal is. The goals can be catch the attacker, analyze the attack, mask the attack from users, sustain service, maximize data integrity, maximize data confidentiality, or minimize cost. The plan also includes one or more plan steps which are techniques for accomplishing a response goal. To accomplish the plan steps the response plan includes methods for executing the plan steps, called tactics. To develop the response plan the analysis agent uses the response taxonomy agent and policy specification agent to apply the dimensions of the Carver Response Taxonomy. When a plan is developed the analysis component monitors the attack, and if necessary adjusts the plan. If incident reports continue to be added to the analysis component it

has to be decided if it should stick to the plan, change it, or build a new plan. Carver has given a thorough explanation of what a plan step includes in [1].

To classify the response goals the analysis components use the services provided by the response taxonomy component. The response taxonomy component implements all dimensions of the response taxonomy, described earlier, except the environment constraint dimension. It uses this taxonomy to determine response weights.

A limitation on response goals is handled by the policy specification component. It also perform a filtering of the plans and tactics generated by the analysis component. Responses restricted by environmental limitations are removed from the response plan in this component. Environmental constraints are set in the system administrator interface. Such a constraint could be certain local governmental laws.

The responses are carried out by the response toolkit component. This module also measure and provides feedback on the success or failure of the execution of the responses. One particular tool in such a kit, could be to suspend a user account.

5.4.2 Evaluation

AAIRS works as a good framework for an Intrusion Response System. What makes it unique is that Carver have contributed to the actual components as well. The framework has all the inner workings in place.

Another strong point is that the interface must be tailored to the input system. This makes it ideal for any IDS to work with the system.

It is based on agents, which means it should be a fairly scalable solution in terms of performance (it can be divided over a cluster of computers). The communication protocol between agents is not specified, so it is entirely up to the implementation.

This system would be categorized as an adaptive response system, see section 3.4.3, which is the top shelf of automated response systems.

A drawback is its taxonomy, see section 3.5.2, which is primarily based on HIDS input. We want to use the system with input from a NIDS.

There is still much to be sought for in terms of artificial intelligence in different parts of the system. Among non-automated components, the interface relies on humans to update its IDS confidence metric. The policy specification agent is also a static component, which does not catch the dynamism of computer networks.

5.5 Choice of IRS

The main reason for including the FlexResp2 module is that is specialized for Snort. It is however just a primitive rule based Intrusion Response System, and it can not compete with the more advanced systems. Snort and FlexResp2 combined does however, fulfil the functions of an IPS most commercial vendors provide.

SARA is an interesting architecture and provides many good thoughts on how to fulfil DARPAs two main hypothesis. It does however, not include specifics on how most of the components should be implemented.

AAIRS is a complete IRS, with the inclusion of how to specifically build the different components in the system. The best part of it is how it has divided the functions into the components, so parts of it can be used as they are. This makes it a great architecture to build other IRSs on.

AAIRS is our top choice of architecture, and so it will make the basis for our own IRS in the following chapters.

5.6 Summary

In this chapter we have looked at three technologies for Intrusion Response Systems; the FlexResp2 module for Snort, the SARA architecture, and Carvers AAIRS.

These technologies have been evaluated, and we have reached a conclusion on what approach we will use to make our own IRS.

In the next chapter, we will use AAIRS as the basis of our own IRS.

6 Proposed Response Taxonomy

6.1 Introduction

The response taxonomy is one of the most important parts of the response system. It is used to evaluate the attack and the exposed system. This is done by classifying the incidents and calculating a weight so the response plan can be assigned a value. The value assigned is the measure on how appropriate the response plan is. In AAIRS, Carver presented a similar taxonomy; we used the main idea and adapted it to NIDS sensors.

6.2 Time of attack

The time of the attack is an important dimension of the response taxonomy, and represent what phase the attack is in. To evaluate which response goals to implement, time is critical. Some responses are not desirable before the main attack has started, and some are not desired when the attack is over. The time of attack dimension is divided in three sub-classes; before the attack (pre), during the attack, and after the attack (post).

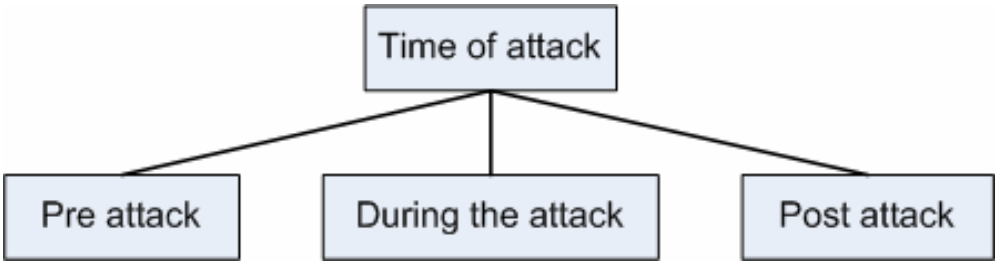

Figure 9: Time of attack taxonomy

6.3 Type of attack

The type of attack is the second dimension of the taxonomy. Different attack types require different response goals. This make it important to decide what type of attack is going on. The determination of type of attack is done by a response taxonomy model. We have chosen to use a taxonomy based on the Lindqvist Taxonomy [13], which is described in section 3.5.

By using the basic ideas of the Lindqvist Taxonomy we have made a proposal to a taxonomy, which is able to divide different types of attacks in different classes.

We have divided the type of attacker into two different sub-classes; intrusions; and denial of services. The intrusion class is attacks intended to reveal or change protected information. This class replaces the exposure class in Lindqvists Taxonomy. In the denial of service class, attacks intended to block services are placed.

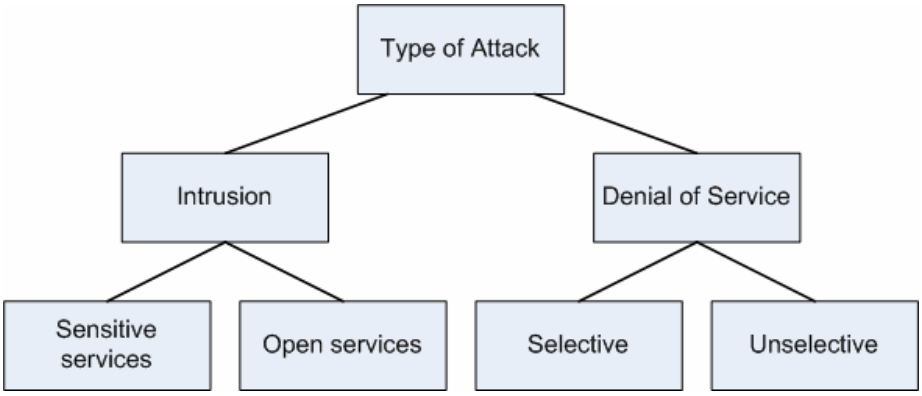

Figure 10: Proposed Taxonomy of Attacks

Intrusion is divided into two subclasses:

- Sensitive services: These are services protected from outside access. Services like backup and other internal administrative functions that should not be accessed from the outside.
- Open services: Attacks at services available to the outside, like WWW, FTP, mail and so on.

Denial of Service is also divided into two subclasses:

- Selective: Attacks to block a specific service.
- Unselective: Attacks that blocks all services on a part of the network or the whole network.

6.4 Type of Attacker

When applying responses to an attack the tactics can change dependent of what type of attacker is attacking the system. An experienced hacker may use combinations of automated scripts to exploit the system in a way an automated attacker is not capable of. How to decide if an attack is performed by a manual or an automated attacker can be almost impossible.

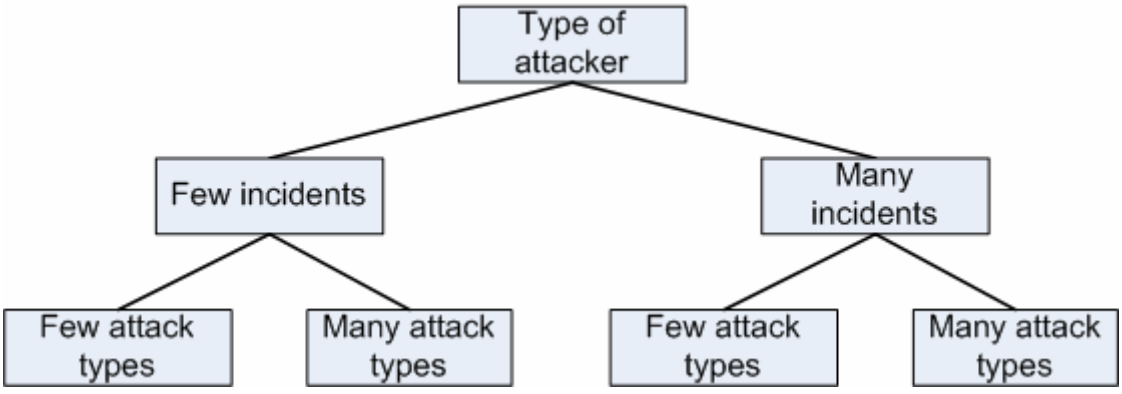

Figure 11: Type of Attacker taxonomy

We have chosen to look at the type of attacker from the attacked system point of view, and divided this dimension up into two sub-classes; many and few incident reports.

These classes are equally divided up in two sub-classes; one containing attacks with several types and one with few.

By dividing attacks in this way equal attack tactics are placed in the same class, independent of the nature of the attacker.

6.5 Strength of suspicion

The strength of suspicion is an attribute saying how certain we can be that the incidents are real attacks. This certainty is calculated by a formula presented by Carver in AAIRS.

```
Suspicion =
(number of Attacks + Number of incidents)/2*IDS confidence
```

The strength of suspicion is calculated by adding the total number of monitored ongoing attacks and the number of incidents added to this specific attack. This sum is divided by two and multiplied by the confidence of the reporting IDS.

A problem with IDS sensors is that they generate false positives, which is quite normal behaviour. This can also be a result of bad configuration and tuning. Some traffic that triggers an alert is not necessary hostile traffic, but can be extreme cases of legal traffic. Every incident delivered from the interface module contains an IDS confidence metric. This is assigned with a value that represents how good the sensor is. By good we mean how few false positives a sensor has. Alerts from reliable IDSs give a higher degree of suspicion. A bad configured ID can lead to great consequence as an input device for a response system. I.e. if amazon.com shuts down the web server, because an IDS has reported false positives, amazon.com can not sell any books.

6.6 Attack implications

This dimension handles the fact that different services in the system are more important than others. This means that some responses are unsuitable on some critical services on the network, and suitable on others. The importance of services is presented by applying a cost value to each resource in the network. By arranging services in different groups, cost can be applied to the whole group. This will make the process on applying costs easier. A higher cost means a more important service.

6.7 Environmental constraints

This is a relatively open dimension. This is where the responses are restricted by law or organizational regulations. Some responses may be preferred before others, some always preferred, and some never preferred. These are choices done by the system administrator.

6.8 Summary

To be able to classify which response to apply in the given situations we have developed our own response taxonomy. The taxonomy includes six different aspects applying a response, called dimensions.

In the next chapter we present our IRS, which utilizes this taxonomy.

7 Conseptual Intrusion Response architecture

7.1 Introduction

In this chapter we will describe our proposal to an automated response architecture. This architecture is based on Carvers AAIRS, which provides an good framework for Intrusion Response Systems.

7.2 Before we begin (an explanation of terms)

7.2.1 Plan step Tactical Implementation (PTI)

A PTI is the representation of a response goal. This goal can be reached by one or several possible plans. The plans consist of one or several response actions. The plans are given a weight so the analysis agent can decide which is the most appropriate response plan.

7.2.2 Incident Report

The Incident Report (IR) class contains all the information on the alert presented by the IDS. The content of the IR is the decision base for the analysis modules and the response taxonomy.

7.3 The components

This section is a presentation of the architecture. It is based entirely on Carvers AAIRS with some essential improvements. The model consists of several modules.

7.3.1 Interface

The interface module is the connection between the NIDS and the IRS. In the module every NIDS has its own plug-in module, responsible for interpreting the NIDS alert to an IR. By making this as modules it is easy to connect both several NIDS and different NIDS. Since there is no single standardized format[1] for an NIDS alert we have decided to propose that every type of NIDS has its own type of module.

Each NIDS has its own interface component. We are replacing Carvers single confidence metric, with an alarm matrix described in [29]. This method is superior, as it does not take into account the general quality of the IDS, but rather how good it is on detecting certain attack types.

[1] There are some proposed standards, but none have been adopted in a large scale nor implemented in most current systems.

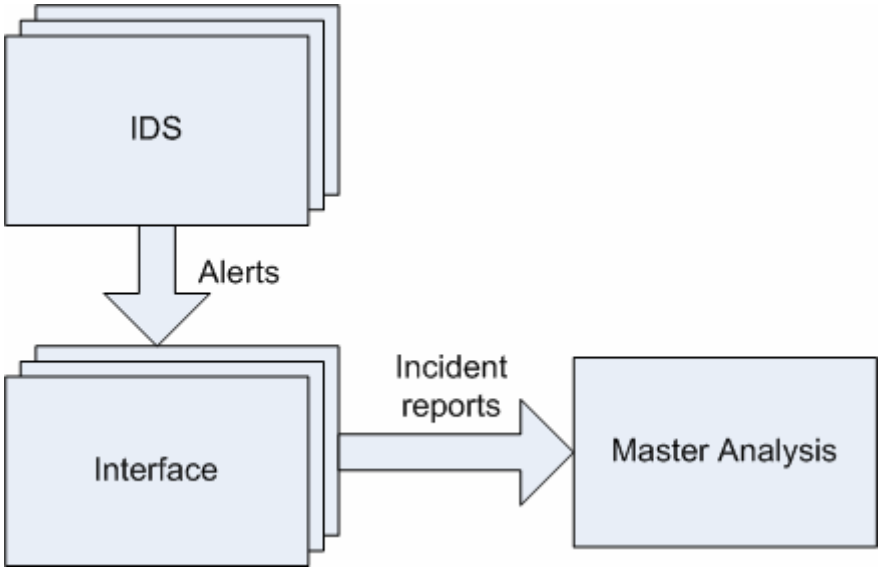

Figure 12: Incident Report from Interface to MA

This information is then added to the incident report and sent to the Master Analysis.

The alarm matrix is already established during an initial setup period, and does not change dynamically.

7.3.2 Master Analysis

The basic principle of the Master Analysis (MA) is the same as in Carver's AAIRS [1]. The MA's object is to determine if an alert is a part of an ongoing attack, or should be threaded as a new attack. The part that differs from AAIRS is the part where the decision is made. We use other criteria for determining if the incoming alert is a part of an ongoing attack, or an alert for a new attack. To determine if an incident is a part of an already monitored attack, the incident time, IP addresses, type of protocol (like TCP, ICMP and so on), and port numbers.

The search

The MA includes a reference list of analysis agents already created. To check if an incident report is a part of an already discovered attack or a new attack, the MA searches through existing Analysis Agents (AA). If no match is made in the search, the incident reported is treated as a new attack. Then a new AA is created for this incident. The values of the AA are used to compare against, are the values of the last added incident.

When the MA searches through the list of AAs it must access the specific AA history list. This list is made up by the incident reports and its attributes. The result of the search is an appropriate AA or that no match is made. If no match is made a new AA is created and added to the MA internal AA list.

Simple determination model

For determining if an incident is a part of an ongoing attack or a new attack, we must look at the network information. Our thesis is that if the source IP and protocol type on alerts are the same and the time of incidents is less than 60 minutes, then the incidents

are a part of the same attack. If an attacker attacks from a specific IP all incidents with this destination IP within 60 minutes is treated as the same attack. This is a simple model, but there are still several obstacles to overcome for the model to work.

In some cases it is the responses from our system that triggers the IDS alarm, not the attacker's action. In these cases the destination IP is the IP of the attacker. To deal with this case we must also see if the destination IP is the same as the source IP of already monitored attacks. By using this solution we also get the opportunity to recognize distributed attacks at a single resource on our network. If a distributed DoS attack is launched against a single resource, all incidents will be handled by the same analysis agent.

Another obstacle is if the attacker is sited behind an ISP and renews his IP address. To handle this scenario attacks coming from the same IP network is seen as the same attack. This is done by comparing the three most significant octets in the IP address. If the ISP is very large, it is a possibility that two different attackers perform an attack at the same target. The probability for this to happen is very small, but to reduce it even more the permitted time gap for incidents belonging to each other could be shorter.

There is always a possibility that some alerts belonging to one analysis agent are sent to another. This is a problem which can not be removed, just minimized.

Refined model

A possible refinement is to introduce a match score. The new incident is compared the same way as before, but instead of just committing it to the first and best AA we search through all AA to find the best. Incidents must fulfil the requirements of the same IP, or more precisely the same IP segment, and the incident must occur within 60 minutes after the last incident. The match score is given by two criteria. A higher value is assigned as shorter the time interval is from the last incident.

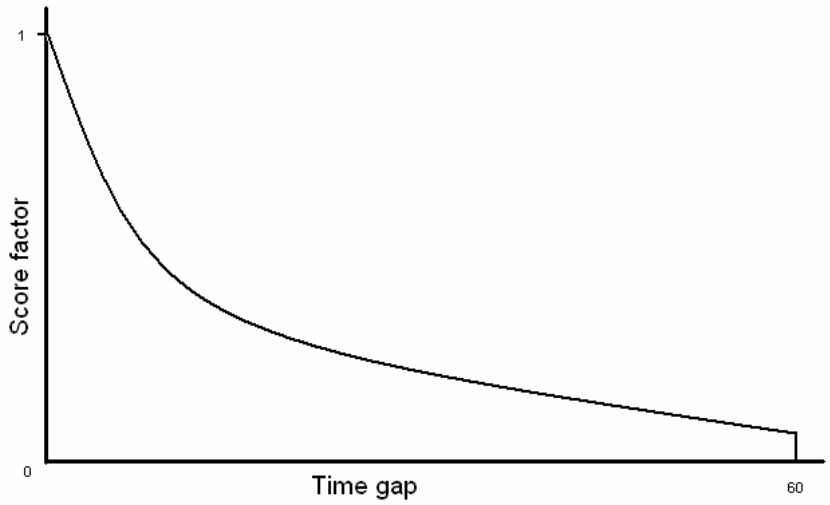

Figure 13: Score factor vs Time gap in minutes

The second criterion is if the source IP is exactly the same IP or just the same IP segment. A higher score is given if the source IP is the same, than if the source IP is located in the same segment.

Every AA that matches the basic criterion is added to a temporary list which also includes the match score achieved. When the search is completed, the AA with the highest match score is selected.

Creation of a new AA

If a received incident report does not match any existing AA, a new AA must be made. The MA must then create a new AA and add it to the MA's internal AA list. The Incident Report is sent to the new AA. The AA is created with a connection to the Response Taxonomy module and the Policy Specification module. If an AA does not receive a new incident before 60 minutes after the latest incident, the AA is deleted from the memory and stored for later analyses.

7.3.3 Analysis Agent

For every attack an Analysis Agent (AA) is created to follow the attack. The AA receives incident reports, which is decided to be incidents of the same attack by the Master Analysis. This incident report is added to the AA's internal history list.

When created the AA builds an initial response plan. The plan is called Plan step Tactical Implementation (PTI). The PTI includes a response goal, which includes one or several plan steps that again can consist of several response actions. The AA uses the Taxonomy Agent to classify the incident report and weight the possible PTIs.

The Response Taxonomy component applies all dimensions, except the environment, and returns a weighted tentative plan. Here, different response goals are weighted. The most appropriate response goals are given the highest weight. Next the goals are constrained by the Policy Specification component.

When returned from the Policy Specification component the final plan is selected. The selected plan is sent to the response toolkit.

7.3.4 Response Taxonomy Agent (RT)

The RT classifies incident reports on the basis of a response plan and weights the tentative response plan. The RT applies all dimensions of the response taxonomy, except the environmental dimension. By applying the dimension of the response taxonomy the RT can classify the IR. Responses that is not feasible for the specific attack is removed from the response plans and the existing response plans are weighted.

7.3.5 Policy Specification

In this module the environmental constraints are applied. There are constraints set by the system administrator. Here the system administrator can set constraints given by i.e. national laws and organisation policies.

7.3.6 Response Toolkit

The response toolkit is an interface out to the different response tools. When a PTI is sent from the Analysis Agent, it is the job of the Response Toolkit to execute the necessary actions. The toolkit includes a list over the available responses. This list holds references to different response tools.

7.4 Summary

We have presented a conceptual architecture of an IRS prepared for Network IDSs in general, and Snort in particular. It is based upon Carvers AAIRS, but with changes and enhancements in most components.

In the next chapter, we have implemented a prototype of a system, with main focus on the Master Analysis component.

8 Prototype of Master Analysis

8.1 Introduction

To visualize parts of the framework, a prototype of the Master Analysis is built. The prototype is developed in Java, and includes the foundation of the framework. In this chapter the modules implemented are described.

8.2 Incident Report

The Incident Report is used internally in the model to represent an alert from an IDS sensor. This report includes many attributes, where some are used by our model. Attributes included in the report is: destination IP, source IP, time, destination port, source port, IDS confidence, alert class and alert priority. In our model we only use the time-, source IP-, destination IP-, IDS confidence- and the alert class attributes.

8.3 Master Analysis

This is the core of our prototype. This class is the module where the incoming incidents are analysed to see if it is an instance of a new attack or an instance of an already monitored attack.

The Master Analysis (MA) class contains a reference to the Response Taxonomy Agent (RT) and the Graphical User Interface (GUI) object. These references are relayed to the Analyse Agents (AAs).

When a new incident report is sent to the MA, it is analysed to determine if the new incident is a part of an already ongoing attack. The MA searches through the existing AAs. If a match is made the new incident is added to the specific AA, else a new AA is created to handle the incident. When a new AA is created the MA adds it to the internal history list. The MA class provides a public function returning the number of AA. The MA includes two functions that determine this, one simple and one a bit more refined. The subject of this prototype is to investigate these two methods.

Alternative 1

This approach is basically to see if the new incident is from the same source and is time stamped within 60 minutes from the last incident. If the incident fulfils this criterion it is treated as a part of the same attack. The approach also handles incidents triggered on a reply to the hacker, and the problem of attackers renewing their IPs.

Alternative 2

In this alternative we use the same attributes, but in a different way. Here we introduce a score system. The existing Analysis Agents is given a score for how good they match the new incident. To be taken into consideration the time gap between the last incident of an AA and the new incident must not be 60 minutes or more. If the AA passes this test, it is given a time score dependent on how small the time gap is. The time score is calculated by the formula: `1/(gap+1)`. Further, this score is multiplied by a factor. If the source address of the last incident added to the AA and the new incident matches, the score is increased. If the IPs are from the same segment more score points are added.

When the AA has been applied a score, the AA id and the score is added to a table. After all AAs are analysed, the new incident is added to the AA with the highest score.

8.4 Analysis Agent

The Analysis Agent class does not completely fulfil the functionality in the framework presented in chapter 7. This simplification is done because the objective of the prototype is to investigate the analysis process in the Master Analysis module. In the prototype, the AA only store the incidents added to it and present the list of incidents and the AA ID number.

A new AA is started in its own thread by the Master Analysis, with the references to the Response Taxonomy and the GUI. It is also given an identification number by the Master Analysis.

When the Analysis Agent receives a new incident report it is added to the history list. The history list can be shown by calling the public function showHistory, which returns an ArrayList of IncidentReports. It also calls a public function in the GUI, called updateAATree, to add the incident to the Analysis Agent in the graphical representation of all AAs.

8.5 Graphical User Interface (GUI)

The GUI is used to launch new incidents and the Incident Tree, which is graphical view of the existing Analysis Agents and the incidents added to them.

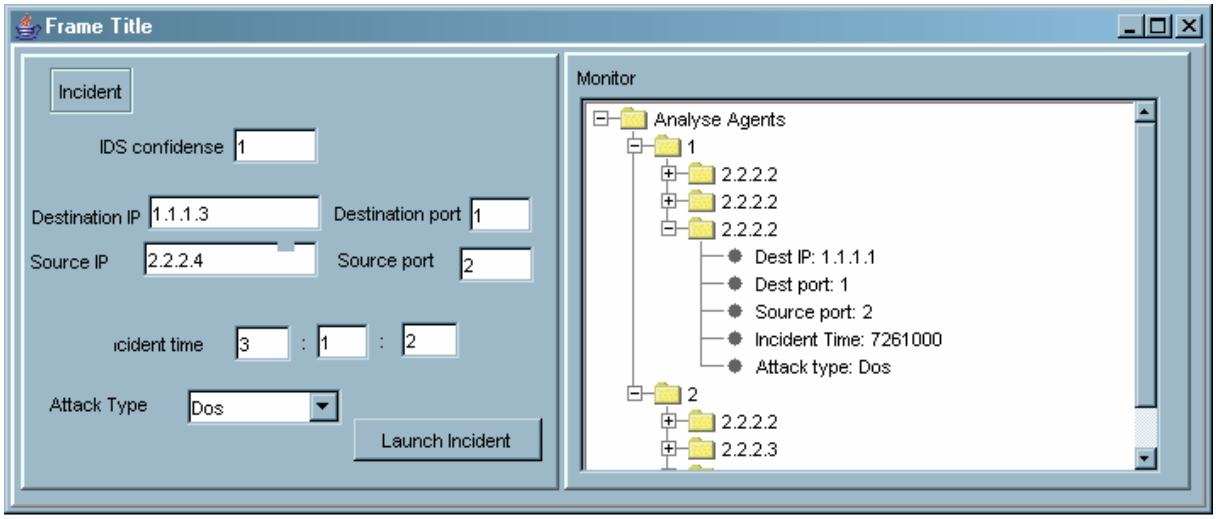

Figure 14: Prototype GUI

Launch new incident

To launch a new incident, the IDS confidence, source IP and port, destination IP and port, attack type, and incident time must be filled in. These are all attributes that should be presented by the IDS.

Incident Tree

All Analysis Agents and the respective incidents are presented graphical in a tree structure. The Analysis Agents is presented with the ID number as children of the root (Analysis Agent). The incidents added to the Analysis Agents are represented as children labelled with the source IP. The leaf nodes are the incident reports characteristics.

8.6 Summary

In this chapter we have described how we have implemented parts of the framework in Java. We have verified the functionality and methods of the enhanced Master Analysis.

To conclude the IRS portion of this thesis, we present our IRS integrated with Telenors IDS structure in the next chapter.

9 Integrating with Telenors IDS

9.1 Introduction

In this chapter we will propose some complete solutions integrating the existing Telenor IDS and our IRS framework. The existing Telenor system is described in appendix B.

The framework we presented in chapter 7 provides in combination with the IDS provided by Telenor a complete IDR. We will suggest two slightly different solutions, one local and one centralized.

9.2 Local solution

The idea behind this solution is that the IRS system is located locally and is working on the local network.

Since we do not want to interfere too much with the existing Telenor solution, we want the IDS to deliver the alerts both to the local database and a unified output module. The unified module logs the alerts to a file, which is read by a Barnyard [Ap. A] module that sends it to the interface component in the IRS. The IRS should be located on a separate host. This should be done to prevent the IRS in restraining the IDS host.

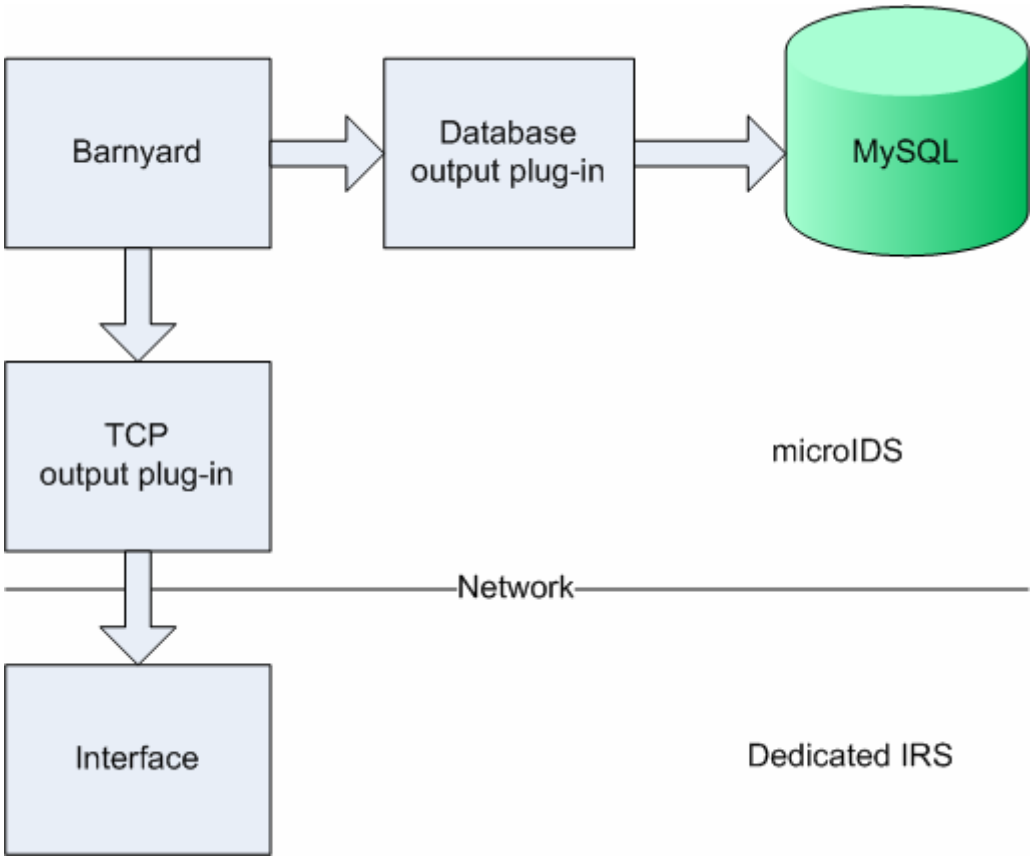

Figure 15: Local IRS connection

By using the Barnyard output plug-in the alerts is forwarded into the interface module of the framework. This solution will result in a very quick transmission of the intrusion

alarm to the interface module. On the other hand by implementing a local solution the advantages of a global view is not present.

Since the localization of the IRS is in the guarded network, it has a wide variety of responses to choose from. It should have privileges to change firewall rules, shut down hosts, and in extreme cases; track down the opponent.

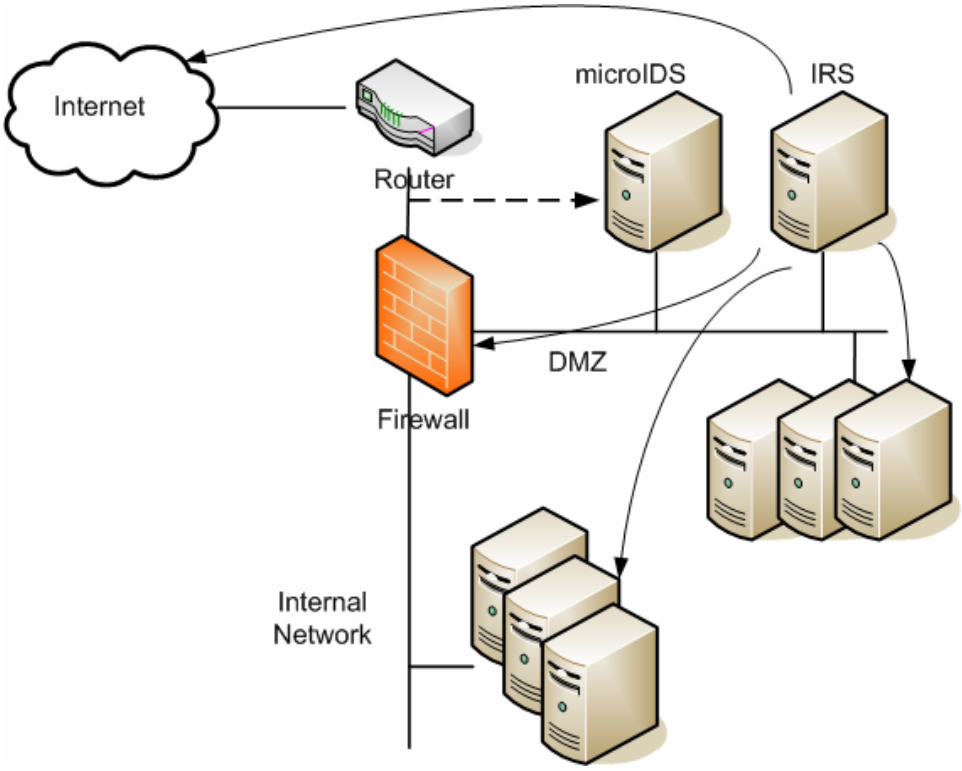

Figure 16: Local IRS controlling the network

9.3 Centralized solution

By connecting the interface module on to the centralized Oracle database, we will provide a global view of many IDSs. This will give a better analysis foundation for the IRS.

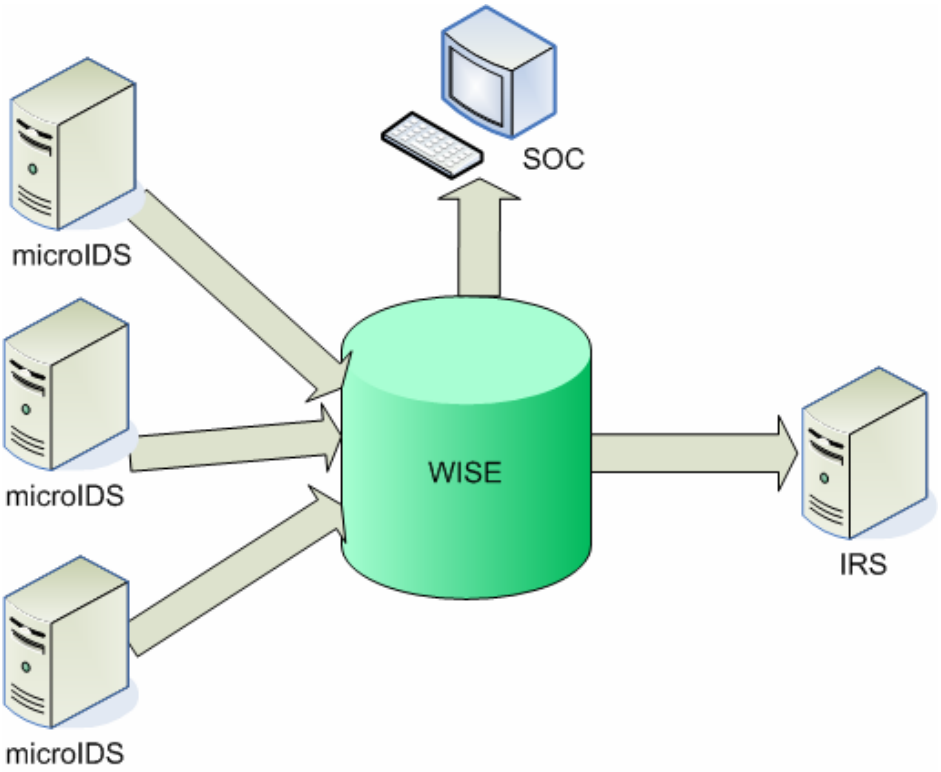

Figure 17: Centralized IRS location

By adding a trigger to the centralized database on incoming alerts, the alerts can be forwarded to the interface module at the same time it is stored in the database. This solution will result in a longer response time, since the alerts are transported by microIDS to the centralized database. Another problem is that the decisions on responses are made by a component outside the guarded network. If configurations in the local network can be set from the outside this will represent a big security risk in it self.

9.4 Summary

In this chapter we have presented several implantations of intrusion response systems on the existing Telenor system with pros and cons. We have described where the IRS will be connected in the existing system and how they will affect each other.

10 Discussion

10.1 Introduction

In this thesis, our work started with the introduction of a finer grained categorization of automatic response systems. We carried on with a literature review, to get the status of the research done in the IRS field, and to see if there are ready-to-run commercial or freeware applications which would work satisfyingly. We then explained in some detail and evaluated three IRSs, before we presented our own response taxonomy. This taxonomy is then used in conjunction with our proposed IRS, which we have partially implemented in a prototype. Finally, we have presented some solutions on how our proposed IRS could be integrated with Telenors existing system.

The purpose of our work has been to gather information related to IRS, and to see if there were applicable systems to complement Telenors existing IDS architecture.

10.2 Evaluated intrusion response systems

FlexResp2, SARA, and AAIRS are stand-alone Intrusion Response Systems. These are all good technologies, but have different approaches on how to solve the response problem.

FlexResp2 has been evaluated because of its close connection with Snort, which are the basis of Telenors microIDS sensor. It does however just utilize the simple method of static association between an alert and a response. This is its greatest weakness, as this could be exploited as mentioned in section 3.4.3.

The FlexResp2 module could however, be used as a part of the response toolkit. It provides methods for breaking down connections, and manipulate ICMP answers to the attacking party.

SARA does not provide any specifics on how to solve the actual intrusion response dilemma, but introduces a framework with an inner and outer loop. This local and global view perspective is a good approach to handle intrusion response and has been adapted to our integrated solution.

AAIRS is an agent-based solution, or if situated in only one place, a system of interchangeable components. It needs however, to be adapted to accept input from a NIDS.

10.3 Taxonomy

In our response taxonomy we used the six dimensions of a response taxonomy proposed by Carver. In the *Time of Attack* dimension we decide which phase the attack is currently in. This is done so we can eliminate responses unsuited for the specific attack phase. When the attack type is classified in the *Type of Attack* dimension it is possible to propose responses handling the specific type of attack; different attack types need different responses. The *Type of Attacker* dimension determines what methods the attacker uses, and by this we can rate the different response tactics. We chose to look at

this dimension from the network point of view. This classification is more accurate since it can be hard to determine the nature of the hacker. The *Degree of suspicion* dimension corrects the inaccuracy in the output from the IDS. The action of applying a response will in almost every case have some consequences, and we need to know how reliable the information is. Another dimension handling consequences is the *Attack implications* dimension. In this dimension we want to address the fact that different resources in the network have different importance for the organization owning the network. To get the ability to set organizational regulations on intrusion response, the *Environmental constraints* dimension is applied.

10.4 The conceptual approach

In the Interface component we are using a confidence matrix. This matrix holds the confidence of every type of attack in one IDS. This is a feature that will make the calculation of the degree of suspicion more precise.

When we developed the framework we needed to adapt the decision base for the Master Analysis. We have proposed that the type of protocol, IP addresses, port numbers, and incident times are the variables is to be used to decide if an incident is an instance of a new attack of an already monitored attack. This is information that is given from the IDS, and is the attributes traditionally used by Telenor to cluster incidents. The simple determination model is a static model that leaves no room for tuning and adjustments. By applying the refined determination model the possibilities to enhance the different attribute matches is preserved.

The fact that the Analysis Agents is agent based the possibility of scaling is close to unlimited. The agents can easily be distributed over several different computers.

The dimensions of the response taxonomy are divided on to the Response Taxonomy Agent and the Policy Specification. By doing this the constant dimensions are handled by the Response Taxonomy Agent, and the more dynamic dimension of Environmental constraints is handled by the Policy Specification module.

By using a component handling the response tools, new response tools can easily be added or removed.

10.5 Integration issues

The local integration solution seems to be the more feasible of the two proposed methods. It has fast response times, and it has a local view of the guarded network it resides in.

The apparent downside to selecting this solution on its own is the missing notion of a global view. With a slightly modified approach to the centralized solution, one could have an instrument to detect larger scale attacks and response to them in a timely manner. This could mean letting administrators know that an attack is going at the neighbour network and it has exploited certain vulnerabilities.

The risk of automated responses is another matter. Telenor have a very strict view on what should be done automatically, and so a manual system could be deployed on top of our IRS.

An implementation of this, would be that the system gives a detailed message where the administrator could choose between a variety of responses. This would on the other hand, prevent the instantaneous response needed in many situations.

10.6 Further work

There are several issues in this thesis that could be subject to further work. Below we have assembled key points of interesting research.

- A good research paper could be written on the topic of correlating intrusion detection alerts from HIDS and NIDS. They would report to their interface, and the master analysis agent should be able to see that alerts correspond with each other. The analysis agent would then use this to strengthen or weaken the alert confidence.

- Human intervention should be avoided as much as possible, so machine learning should be implemented in several components. A study of what algorithms, and in which parts they would have the best impact would be valuable research.

- Enhancements should be done to the policy specification, or a component should be added which modelled the guarded network. It would be interesting to implement the work done in [31] or [32].

- Interesting research would be the social and legal issues involved. There are many responses, like counter-attacking the source, which are seen as a grey zone. A clarification on such issues would be interesting for the non-technical reader as well.

- Last, but not least, a good bachelor thesis would be to implement the proposed IRS, and do tests with real intrusion detection data.

11 Conclusion

This thesis has been focused on a specific research area in the computer and network security field; Intrusion Response Systems. A status has been made on the current situation, and historical events have been established.

We have also partially discounted the rather simple commercial approach of integrated Intrusion Prevention Systems. These systems are all categorized as association based response systems, and these are the most basic type of automated response systems.

During the literature review, a couple of interesting approaches crystallized themselves. After an evaluation of these systems, one system was found to have the properties we needed to propose our own Intrusion Response System.

Our proposed IRS is built upon AAIRS, which is an agent-based and dissectible architecture which components can be enhanced and replaced. We have adapted the system so it is suitable for use with alerts from Network Intrusion Detection Systems and proposed several improvements on the existing system.

Improvements has been made in the interface component of the system, where the confidence metric has been replaced a more sophisticated alarm matrix. This matrix provides a more precise indication on how good the given IDS is to detect a specific type of attack. We have also provided a method to determine if an alert is a new attack or a continuation of an already registered attack based on information from a NIDS. This method corresponds with how it is traditionally done manually at Telenor Sikkerhetssystemer.

To make the transition from a host-based architecture to network-based architecture, a different response taxonomy has to be provided as well, and we have made a suggestion on how such a taxonomy should look like. Our main contribution is a new determination method to identify different attack tactics.

To integrate our IRS with Telenors systems, minor changes has to be done at Telenor. Their installed base of microIDS sensors have to be upgraded to suit a local placement of an IRS. We are confident however, that such a system is possible to deploy in a relatively short time period.

With this proposed solution, swift responses can be executed in a timely matter without much involvement from humans.

Abbreviations

AA	Analysis Agent
AAIRS	Adaptive Agent-based Intrusion Response System
ACK	Acknowledge. Used as a reply message on a TCP message.
DMZ	De-Militarized zone
GPL	GNU General Public License
HIDS	Host-based IDS
IDR	Intrusion Detection & Response system
IDS	Intrusion Detection System
IRS	Intrusion Response System
ISP	Internet Service Provider
MA	Master Analysis
NIDS	Network-based Intrusion Detection System
PTI	Plan step Tactical Implementation
RT	Response Taxonomy agent
RTI	Real Time Infrastructure
SDAR	Short for; Sensor, detector, arbitrator, and responders

Definitions

Alert Alarm and alert is used interchangeably.

Incident Same as alert. Called incident in the IRS.

Attack An intrusion attempt consisting of one or several incidents.

False positive The term “false positive” is the occurrence of an alert being raised when there in fact is no attack going on. An IDS should have a healthy amount of false positives to ensure no attacks go unnoticed.

False negative The case where an attack is not detected is called a false negative, meaning no alert is raised even though it should have been. This is not an acceptable situation and the IDS should therefore be fine tuned to generate more false positives instead. Handling one alert to many is much easier to handle than missing an attack.

References

[1] C.A. Carver, "Adaptive Agent-Based Intrusion Response" *Ph.D. Dissertation*. Department of Computer Science, Texas A&M University, College Station, TX, 2001.

[2] P.A. Porras and P.G. Neumann, "EMERALD: Event Monitoring Enabling Responses to Anomalous Disturbances" in *Proc. 20th National Information Systems Security Conference*, Baltimore, MD, October 7-10, 1997, pp. 353-365.

[3] P.G. Neumann and P.A. Porras, "Experience with EMERALD to Date" in *Proc. 1st USENIX Workshop on Intrusion Detection and Network Monitoring*, Santa Clara, CA, April 11-12, 1999. Available at http://www2.csl.sri.com/emerald/downloads.html.

[4] E.A. Fisch, "Intrusion Damage Control and Assessment: A Taxonomy and Implementation of Automated Responses to Intrusive Behavior" *Ph.D. Dissertation*, Department of Computer Science, Texas A&M University, College Station, TX, 1996.

[5] G.B. White, E.A. Fisch, and U.W. Pooch, "Cooperating Security Managers: A Peer-based Intrusion Detection System" *IEEE Network*, vol. 10, no. 1, January/February, 1996, pp. 20-23.

[6] K. Hwang, S. Tanachaiwiwat, and P. Dave, "Proactive Intrusion Defense Against DDoS Flooding Attacks: Adaptive Filtering with Security Datamining – The NetShield Approach at USC", submitted for review by *IEEE Security and Privacy Magazine*, April 14, 2003, University of Southern California, CA.

[7] J. Koziol, "Intrusion Detection with Snort", *Sams Publishing*, 2003.

[8] J.P. Anderson, "Computer Security Threat Monitoring and Surveillance", *Tech Report*, April 15., 1980, 9F296400 J.P. Anderson co. Fort Washington

[9] D.E. Denning, "An Intrusion-Detection Model", *IEEE Transaction on software engineering*, vol. se-13, no. 2, February, 1987, pp. 222-232.

[10] E.E. Schultz and R. Shumway, "Incident Response – A Strategic Guide to Handling System and Network Security Breaches", *New Riders Publishing*, 2002.

[11] K.-P. Kossakowski et al, "Responding to Intrusions", *Carnegie Mellon Software Engineering Institute (CERT)*, February 1999.

[12] T. Toth, "Improving Intrusion Detection Systems", Ph.D. Dissertation, Institut für Informationssysteme, Technischen Universität Wien, May 2003.

[13] U. Lindquist and E. Jonsson, “How to systematically classify computer security intrusions” in *Proc. 1997 IEEE Symp. On Security and Privacy*, Oakland. CA. May 4-7, 1997, pp. 154- 163.

[14] W. Lee et al, “Toward Cost-Sensitive Modeling for Intrusion Detection and Response”, *Journal of Computer Security*, vol. 10, numbers 1,2, 2002.

[15] CERT/CC Statistics 1988-2003, http://www.cert.org/stats/cert_stats.html, last accessed May 11, 2004.

[16] CERT CC 2003 Annual Report, http://www.cert.org/annual_rpts/cert_rpt_03.html, last accessed May 11, 2004.

[17] F. Cohen, “Simulating Cyber Attacks, Defenses, and Consequences”, http://all.net/journal/ntb/simulate/simulate.html, last accessed May 11, 2004.

[18] N. Negroponte, “Being Digital”, *New York: Alfred A. Knopf*, 1995.

[19] MIT Lincoln Laboratory, http://www.ll.mit.edu/IST/ideval/data/data_index.html, last accessed May 25, 2004.

[20] M. Thottan and Chuanyi Ji, “Anomaly detection in IP networks”, *IEEE Transactions on Signal Processing*, vol. 51, issue 8, Aug. 2003, pp. 2191-2204.

[21] L. Mé and C. Michel, “Intrusion Detection: A Bibliography”, *Technical Report*, September 2001.

[22] Snort Flexible Response Version 2, http://cerberus.sourcefire.com/~jeff/archives/snort/sp_respond2/, last accessed May 26, 2004.

[23] Mechanisms to Implement Intrusion Response, http://www.sdsc.edu/DOCT/Publications/e2/e2.html, San Diego Supercomputer Center, last accessed May 26, 2004.

[24] M. Castells, ”The Rise of the Network Society”, 2. edition, *Oxford: Blackwell*, 2000.

[25] C.A. Carver, ”Intrusion Response Systems: A Survey”, Department of Computer Science, Texas A&M University, College Station, TX, 2000.

[26] S.M. Lewandowski et al, “SARA: Survivable Autonomic Response Architecture”, in *Proc. DARPA Information Survivability Conference & Exposition II*, 2001.

[27] M. Burgess, “Computer immunology”, Centre of Science and Technology, Oslo College, 1998.

[28] S. Fenet and S. Hassas, “A distributed Intrusion Detection and Response System based on mobile autonomous agents using social insects communication paradigm”, *Elsevier Science B.V*, 2001

[29] S. Tanachaiwiwat, K. Hwang, and Y. Chen, “Adaptive Intrusion Response to Minimize Risk over Multiple Network Attacks”, University of Southern California, 2002. (Submitted for review by ACM Transactions on Information and System Security)

[30] W. Lee et al, “Towards Cost-Sensitive Modeling for Intrusion Detection and Response”, in *Workshop on Intrusion Detection and Prevention, 7th ACM Conference on Computer Security*, Athens, November 2000.

[31] T. Toth and C. Kruegel, “Evaluating the Impact of Automated Intrusion Response Mechanisms”, in *Proc. Of the 18th Annual Computer Security Applications Conference*, 2002

[32] Yu-Sung Wu et al., “ADEPTS: Adaptive Intrusion Containment and Response using Attack Graphs in an E-Commerce Environment”, School of Electrical & Computer Engineering, Purdue University, 2003

[33] TippingPoint Technologies, http://www.tippingpoint.com/products.html, last accessed May 30, 2004.

[34] StillSecure, BorderGuard, http://www.stillsecure.com/products/bg/, last accessed May 30, 2004.

[35] Network Associates Technology, IntruShield White Paper, available at http://www.nai.com/us/_tier2/products/_media/sniffer/ds_intrushieldidssensor.pdf, last accessed May 30, 2004.

[36] Hogwash, http://hogwash.sourceforge.net/oldindex.html, last accessed May 30, 2004.

[37] Snort_inline, http://sourceforge.net/projects/snort-inline/, last accessed May 30, 2004.

Appendix

A Snort

B Telenor microIDS & WISE

C Prototype UML class & sequence diagram

D Prototype JavaDoc

E Prototype code