2012

A Study Of Methodologies Used In Intrusion Detection And Prevention Systems

David Mudzingwa

North Carolina Agricultural and Technical State University

A Study of Methodologies Used in Intrusion Detection and Prevention Systems

David Mudzingwa

North Carolina A&T State University

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Department: Electronics, Computer and Information Technology

Major: Information Technology

Major Professor: Dr. Rajeev Agrawal

Greensboro, North Carolina

2012

School of Graduate Studies

North Carolina Agricultural and Technical State University

This is to certify that the Master's Thesis of

David Mudzingwa

has met the thesis requirements of

North Carolina Agricultural and Technical State University

Greensboro, North Carolina

2012

Approved by:

Dr. Rajeev Agrawal Major Professor	Dr. Ibraheem Kateeb Committee Member
Dr. Naser El-Bathy Committee Member	Dr. Clay Gloster, Jr. Department Chairperson

Dr. Sanjiv Sarin
Associate Vice Chancellor
for Research and Graduate Dean

Biographical Sketch

David Mudzingwa received an Associate degree in Information Technology from Durham Technical College in 2003 and the Bachelor of Science degree in Interdisciplinary Studies from Winston-Salem State University in 2010. He is a candidate for the Master of Science degree in Information Technology.

Acknowledgements

I would like to express my sincere gratitude to my advisor, Dr. Rajeev Agrawal for his guidance, understanding and most importantly his patience and encouragement. I was very fortunate to have an advisor with such immense knowledge to guide me through my research. I am thankful for the freedom and advice he gave me throughout my graduate studies.

I am also thankful to the department Chair, Dr. Clay Gloster, my graduate committee Dr. Ibraheem Kateeb and Dr. Naser El-Bathy for their guidance and help.

I am also grateful to Nicole Fontaine for all her help. Her extensive knowledge of the administrative processes is greatly appreciated.

I would like to express my gratitude to all the kind people around me for their support and encouragement throughout the years.

I would like to thank Bob Toler, Jody Atkins, Carl Lindahl, and Sotorn Muangmanee for the countless discussions and help on related topics that helped me have a better understanding of this area.

Most importantly, this thesis would have not been possible without the love, patience, understanding, and support of my wife Mercy, my daughters Amanda and Ariana and the rest of my family to whom I express my sincere gratitude.

Table of Contents

List of Figures

List of Tables

Abstract

The increase in the security breach of computer systems and computer networks has led to the increase in the number of security tools that seek to protect these assets. Among these tools are intrusion detection and prevention systems (IDPS). IDPS are security systems that are used to detect and prevent security threats to computer systems and computer networks. These systems are configured to detect and respond to security threats automatically, thereby reducing the risk to monitored computers and networks. Intrusion detection and prevention systems use different methodologies such as signature based, anomaly based, stateful protocol analysis, and a hybrid system that combines some or all of the other systems to detect and respond to security threats. Intrusion detection and prevention system comes as an appliance or a software tool. The combinations of the methodologies, delivery mechanisms, and the technical requirements for properly configuring these systems make it difficult to understand and evaluate these systems. This problem is amplified by the lack of publicly available work and current data sets for use in evaluating the effectiveness of intrusion detection and prevention systems. This thesis offers a solution to this problem in three stages. The first stage will offer a clear explanation of the detection methodologies used by the IDPSs and offer a way to compare these methodologies. The second stage will focus on setting up test environments for evaluating both hardware and software based IDPS using publicly available open source tools Tomahawk and Wireshark. The third stage will offer an analysis of the experiments that we conducted using the information presented in the first and second stage and also produce current data sets.

CHAPTER 1

Introduction

Intrusion detection and prevention systems (IDPS) have become a valuable tool in keeping information systems secure. IDPS are security tools that are used to monitor, analyze, and respond to possible security violations against computer and network systems. Although the use and dependency of these systems continue to grow, there are no publicly available ways to evaluate the effectiveness of these systems. The available commercial tools are expensive and are not feasible in some cases. There is also a lack of current work on this problem and the available work and data sets are dated. IDPSs lack an established testing and evaluating processes such as those that exists in the software development field. Unlike the software testing field that has a number of proven tools that available for testing, the IDPS users does not have the same opportunities or the necessary tools to test the IPDS products once deployed. IDPS products also work and behave differently using proprietary rules sets and user interfaces which makes it even harder to evaluate them side by side. Most accuracy and performance metrics on IDPS products tend to be available without the raw data on how the results were produced. Also these numbers are based on lab environments which are not identical to the production environment where the IDPS product will protect. Although these test results are accurate, our research found that most production environments are not identical and that security priorities vary from one organization to the other. We also discovered that a considerable amount of resources are required to properly deploy, run, and maintain an IDPS. IDPS products also use different proprietary detection engines which makes it difficult to evaluate and understand their underlining methodologies. Most of the research work on this issue tends to focus on improving one methodology or evaluating one methodology against a proposed new one.

The objective of this thesis is not to improve any of the IDPS methodologies or propose a new methodology, instead it is to clear the misunderstandings about evaluating IDPS effectiveness by offering a simple but effective way to understand and evaluate IDPS methodologies and products. These problems are broken into three phases:

1. The first phase offers a detailed overview of the four commonly used IDPS methodologies and a simple way to evaluate these methodologies.
2. The second phase explains four ways to setup test environments for evaluating IDPS products using open source utilities and some publicly available evaluation copies of commercial tools. Using the evaluation parameters established during the first phase.
3. During the third phase we will use an advanced setup to run experiments using four IDPS products, our evaluation parameters, and then analyze the results.

The results from our test IDPS products will be evaluated and interpreted as a whole system for example, the performance and overhead are looked at closely since they are interdependent. We also combined easy to use and configuration to gauge how difficult it is to setup, use and manage an IDPS. These are the most important evaluation parameters and in we found that the IDPS products that require less resources to configure and use had a favorable recommendation and their users kept them updated which in turn improved the their ratings.

CHAPTER 2

Literature Review

2.1 IDPS Detection Methodologies

This chapter offers a brief overview of other research work on IDPS methodologies and the evaluations of the IDPS products.

2.1.1 Anomaly based methodology. Intrusion detection and prevention systems are a combination of intrusion detection systems and intrusion prevention systems. Intrusion prevention came out of research on the short comings of intrusion detection. Intrusion detection evolved out of a report that proposed a threat model [9]. This report laid down the foundation of intrusion detection systems by presenting a model for identifying abnormal behavior in computer systems. This model broke down threats into three groups, external penetrations, internal penetrations, and misuse. The report used these three groups of threats to develop an anomaly based user behavior monitoring system. In 1987, "a model for a real-time intrusion-detection expert system that aims to detect a wide range of security violations ranging from attempted break-ins by outsiders to system penetrations and abuses by insiders" was proposed [10]. This model was based on the idea that security breaches to any system can be identified and monitored by analyzing the system's audit logs. The model was comprised of profiles, metrics, statistical models, and rules for analyzing the logs. This model provide "a framework for a general-purpose intrusion-detection system expert system" that is still in use today [11]. Anomaly detection methodologies are plagued with high rates of false positives and a new detection system for anomaly based methodology that strikes a balance between generalizations is proposed [21]. The proposed system balances the generalizations in anomaly detection methodologies and in doing so it achieves both a high accuracy rate and a low false positive rate.

In [25] data mining techniques that are used in anomaly based intrusion detection are explored. A further discussion of the statistical based anomaly methodology is covered in [26].

2.1.2 Signature based methodology. In [17] a structured approach to intrusion detection systems by defining and classifying the components of an IDS system is offered. This classification offered a clear understanding of all the parts that make up intrusion detection systems and the challenges the systems faces. James and Jay offered survey of where the current research is on the techniques and methodologies used in intrusion detection [18]. Their focus was to summarize the research done in intrusion detection to this point and in so doing offered a starting point for future research to start. A technical overview of intrusion detection systems starting with the fundamentals of how these systems are structured to the techniques they use to detect and identify potential security threats are discussed in [19]. This paper also explains how an intrusion detection system responds to violations of the security policies they are monitoring. In a proposal for a new signature based intrusion detection and prevention system [23], the authors started by presenting the basic organization and implementations of intrusion detection and prevention systems. Then they went on to proposed and design a new signature based intrusion detection and prevention system called HawkEye. The authors also compared it with current intrusion detection and prevention systems on the market. In [27] Snort, the most popular signature based IDS is discussed.

2.1.3 Stateful protocol analysis based methodology. Intrusion detection and prevention systems suffer from scalable and efficiency problems, these two problems are addressed by high performance deep packet pre-filtering and memory efficient technique [20]. This technique allows the Intrusion detection and prevention systems to have high accuracy rates and high performance numbers by utilizing a deep packet pre-filter and changing how it handles and processes memory and captured data. In [31] a network based intrusion detection system that is based on dynamic application layer protocol analysis. Protocol analysis is detailed in [32], which combines it with another methodology. In [33] stateful protocol analysis is used as the based for a proposed Web IDS. A new detection engine that is based on understanding the protocols is proposed in [51]

2.1.4 Hybrid based methodology. Combining the two most used methodologies in intrusion detection and prevention systems into a system that uses both anomaly and signature based detection methodologies produces a better detection system [22]. This combination of methodologies produces a better system by preprocessing the data with the anomaly detection engine and then passing the results to the signature based engine. This results in a very high accuracy rate and very low false positives. The two main methodologies used in intrusion detection and prevention systems are combined to form a collaborative intelligent intrusion detection system (CIIDS) [16]. This work looked and addressed current challenges to collaborative intrusion detection systems and the algorithms they employ for alert correlation. It also suggested ways to reduce false positives while improving the detection accuracy. Fuzzy logic and data mining is combined in [28] to produce a hybrid methodology that combines anomaly and signature methodologies. In [29] another combination of anomaly and signature based methodologies is covered. In [30] a new hybrid intrusion detection system is proposed for clustered wireless sensor networks. A new hybrid system for mobile adhoc networks is proposed in [52] and this system combines anomaly and the new system that is based on how the system responds to an attack.

2.2 IDPS Evaluations

The first research that looked the claims of the intrusion detection systems appeared in

1998. This work put forth a frame work for thoroughly testing an intrusion detection system and also offers data sets to be used in the evaluations. This early work is evaluated and checked for accuracy in [34. This work challenges the continual use of the data sets produced in IDPS tests. This work puts on argument that these data sets are out dated and that the procedures used to generate the data sets were not representative of a production network [35]. Another evaluation was conducted that looked at the accuracy of the data sets [36]. A study of the advantages and disadvantages of intrusion detection systems (IDS) performance tests were studied and the resulting work produced a frame work for evaluating IDSs [37]. These limitations in the evaluation of IDPSs were addressed by the Lincoln Adaptable Real time Information Assurance Test bed (LARIAT) [37]. This was a better testing application that used a graphical user interface instead of the command line and was easier to use but was only available to the United States government [37]. Trident evaluation was another work that tried to improve on the early works by introducing ways to add new background and attack traffic to the test data sets [38. This work offered a way to account for evasion techniques during an evaluation of an IDPS. A comparison of two IDPS methodologies to evaluate them is given in [39]. This work describes how to set up a test bed that can be used to evaluate an IDPS using a Snort and Spade. Network captured packets that are used for evaluating an IDPS have to be clean and complete. Cleaning these files and making them suitable for use in IDPS testing is presented in [45]. Before an IDPS is evaluated for effectiveness, its underlying detection methodologies need to be understood. An IDPS can be based on any of the four main detection methodologies. The signature based, anomaly based, stateful protocol analysis based, and the hybrid based detection methodologies are described in [46]. In [49] the effects of regular network traffic on how an IDPS are examined. The first research that looked the claims of the intrusion detection systems appeared in the shortcomings of current IDPS evaluations and vendor claims are highlighted in [53].

CHAPTER 3

Intrusion Detection and Prevention Systems Detection Methodologies

3.1 Introduction

Intrusion detection systems (IDS) are security tools that are used to monitor and analyze computer and network systems for possible security violations. These violations can be a result of break-in attempts by unauthorized external intruders trying to compromise the system or internal privileged users misusing their authority. Intrusion detection systems tend to be passive and only monitor and notifies the administrator of possible threats to monitored systems. Intrusion prevention system (IPS) is an intrusion detection system that has the capacity to prevent intrusions [1, 2]. Both the Intrusion detection system and the Intrusion prevention system can be a software tool or an appliance. The intrusion detection and prevention system (IDPS) is a combination of the intrusion detection and the intrusion prevention systems. All of these systems use the same underlining methodologies to detect and analyze intrusions. The main detection methodologies are signature based, anomaly based, Stateful protocol analysis based, and hybrid based. Figure 1.1 shows a general purpose architecture of which IDPS systems are based. This architecture was developed by the Intrusion Detection Working Group and has four functional blocks, the Event blocks which are the event boxes that gathers events from the monitored system and will be analyzed by other blocks, then the Database blocks which are the database boxes which stores the events from the Event blocks, then the Analysis blocks that processes the events and sends an alert, and final the Response blocks whose purpose is to respond to an intrusion and stop it [3]. The signature based methodology is also referred to as misuse based or rule based [3]. The signature based system works by comparing observed traffic and application behavior to know malicious signatures on in its database.

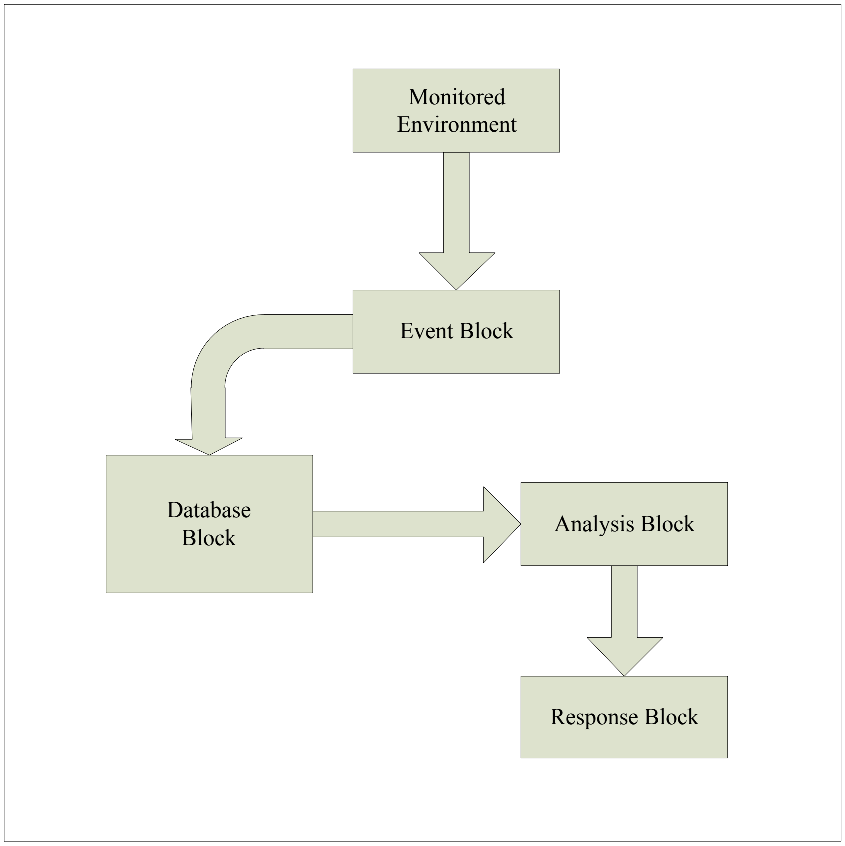

Figure 1.1. General architecture for IDPS systems.

The signature based system is fast and highly effective on known violations or threats, but does not work so well on new threats or variants of known threat. The anomaly based systems works by learning the monitored system of which it builds a baseline profile which it uses as a template and compare against observed activities and any diversions are marked as an anomaly [5]. Anomaly based systems places the most overhead on the systems it is monitoring. Its advantage is that it can detect new threats without any new updates. Stateful protocol analysis methodology works by comparing established profiles of how protocols should behave against the observed behavior. A hybrid methodology works by combining two or more of the other methodologies. All these methodologies are plagued with false positive and false negatives and the anomaly based methodology IDPS produces the most. False positives are when non-security threats are flagged as threats and false negative result when actual security threats are flagged as non-security threats. Some current IDPS products on the market use a hybrid of these methodologies in an effort to be more accurate and reduce the number of false positives and negatives. One way to reduce false positive and false negative is to use alert correlation. Most of the research work done on intrusion detection and prevention systems (IDPS) mainly focuses on defining and detailing one methodology or technology or suggesting a new methodology or technology but stops short of offering a guideline or techniques for comparing all established methodologies, technologies, and systems that use these methodologies. In this thesis, we offer a clear presentation of intrusion detection and prevention systems, starting by defining the four mainly used methodologies, the signature based, anomaly based, stateful protocol analysis based, hybrid based and then offer a way to compare and test IDPS systems and their underlining methodologies. Then conclude by offering a list of current popular IDPS products on the market.

3.2 IDPS Methodologies

There are many different methodologies used by IDPS to detect changes on the systems they monitor. These changes can be external attacks or misuse by internal personnel. Here, we describe the four methodologies in detail. As shown in Figure 1.2, all the methodologies use the same general model and the differences among them is mainly on how they process information gathered from the monitored environment to determine if a violation of the set policy has occurred.

3.2.1 Anomaly based methodology. Anomaly based methodology works by comparing observed activity against a baseline profile. The baseline profile is the learned normal behavior of the monitored system and is developed during the learning period were the IDPS learns the environment and develops a normal profile of the monitored system. This environment can be networks, users, systems and so on.

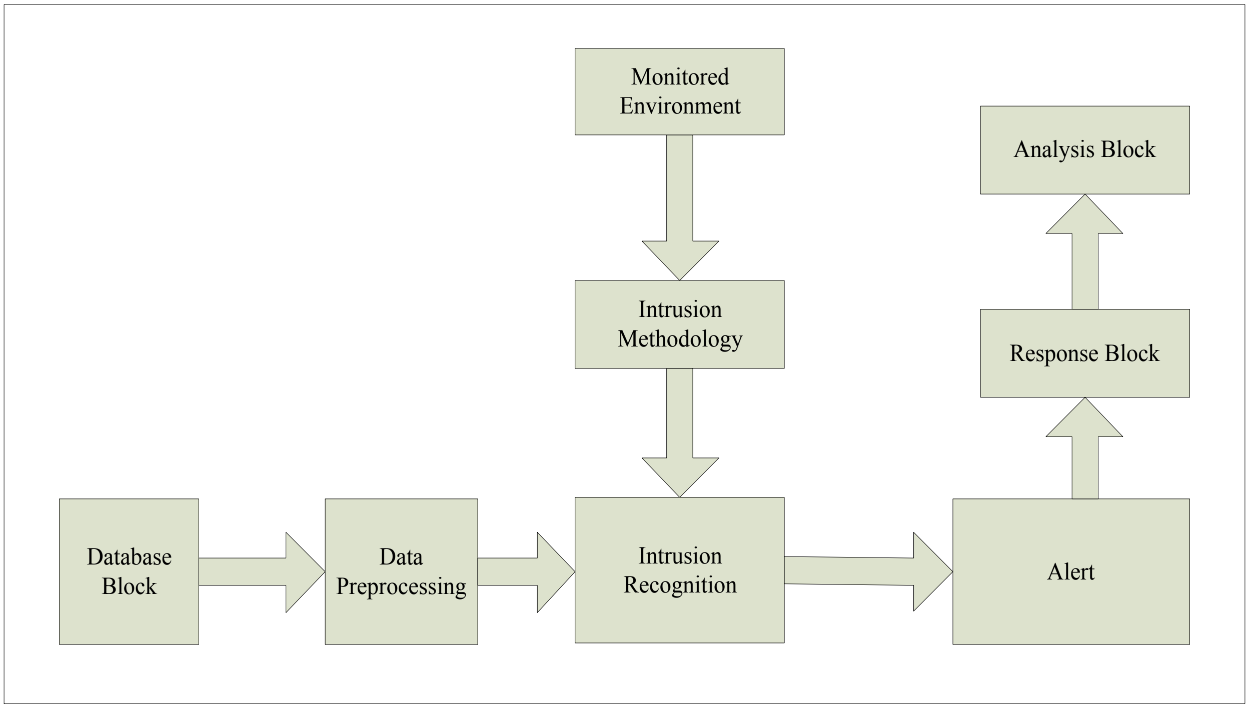

Figure 1.2. Organization of a generalized intrusion detection and prevention system [14].

The profile can be fixed or dynamic. A fixed profile does not change once established while a dynamic profile changes as the systems been monitored evolves [8]. A dynamic profile adds extra over head to the system as the IDPS continues to update the profile which also opens it to evasion. An attacker can evade the IDPS that uses a dynamic profile by spreading the attack over a long time period. In doing so, her attack becomes part of the profile as the IDPS incorporates her changes into the profile as normal system changes. Once the baseline profile is developed and current profile is also created and compared to the baseline profile. Using a predefined threshold any deviations that fall outside the threshold are reported as violations. A new dynamic anomaly detection technique is proposed that uses hidden Markov model for modeling the normal behavior of program through analyzing system calls [24]. The proposed technique uses a two layer detection scheme to reduce false positives and improve detection rates. A fixed profile is very effective at detecting new attacks since any change from normal behavior is classified as an anomaly.

Anomaly based methodologies can detect zero-day attacks to environment without any updates to the system. Anomaly intrusion detection methodology uses three general techniques for detecting anomalies and these are the statistical anomaly detection, Knowledge/data-mining, and machine learning based [8].

The statistical anomaly techniques are used to build the two required profiles, one during the learning phase which is then used as the baseline profile and the current profile which is compared to the baseline profile and any differences that found a marked as anomalies depending on the threshold settings of the monitored environment [1]. Environments that use high thresholds have a higher rate of false positives and those that uses lower threshold might experience a high rate of false negatives. The threshold must be tuned according to the requirements and behavior of the environment being monitored for the systems to be effective. The knowledge/data-mining technique is used to automate the way the technique monitor searches for anomalies and this process places a very high overheard on the system. The technique produces the most false positives and false negatives are produced due to the high overhead that result from the complicated task of identifying and correctly categorizing observed events on the system [4]. The machine learning technique works by analyzing the system calls and it is the widely used technique [7, 14]. The general architecture of an anomaly based IDPS system is shown in Figure 1.3. This architecture is utilized by all techniques that use the anomaly based methodology. As shown in Figure 1.3, the monitored environment is monitored by the detector that examines the observed events against the baseline profile. If the observed events match the baseline, no action is taken, but if it does not match the baseline profile is within the acceptable threshold range then the profile is updated.

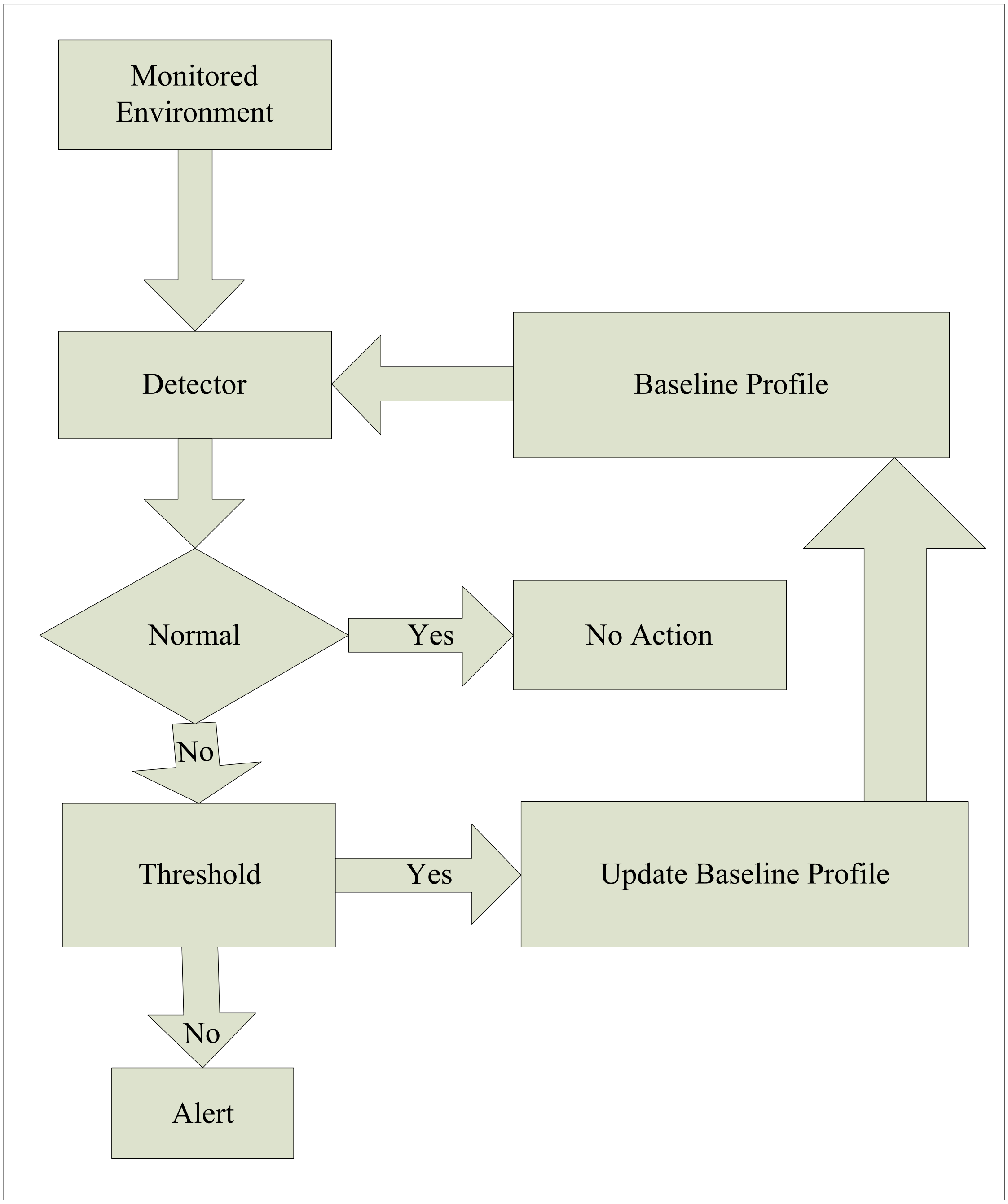

Figure 1.3. Anomaly based methodology architecture.

If the observed events do not match the baseline profile and fall outside the threshold range they are marked as an anomaly and an alert is issued.

The anomaly based methodologies have the following advantages and disadvantages:

Advantages

- The ability to detect new attacks/violations without updates.
- Detects insider attacks
- Detects variants of know attacks
- Does not need signature updates to detect new threats
- Can detect threats that utilize multiple but separate attacks

Disadvantages

- High volume of false positives and false negatives
- Needs a training period before use
- Places high overhead on the system
- Difficult to use due to high volume of alerts

3.2.2 Signature based methodology. Signature based methodology works by comparing observed signatures to the signatures on file. This file can be database or a list of known attack signatures. Any signature observed on the monitored environment that matches the signatures on file is flagged as a violation of the security policy or as an attack. The signature based IDPS has little overhead since it does not inspect every activity or network traffic on the monitored environment. Instead it only searches for known signatures in the database or file. Unlike the anomaly based methodology, the signature based methodology system is easy to deploy since it does not need to learn the environment [4]. This methodology works by simply searching, inspecting, and comparing the contents of captured network packets for known threats signatures. It also compares behavior signatures against allowed behavior signatures. The general architecture of a signature based methodology is shown in Figure 1.4. This architecture uses the detector to find and compare activity signatures found in the monitored environment to the known signatures in the signature database. If a match is found, an alert is issued and there is no match the detector does nothing.

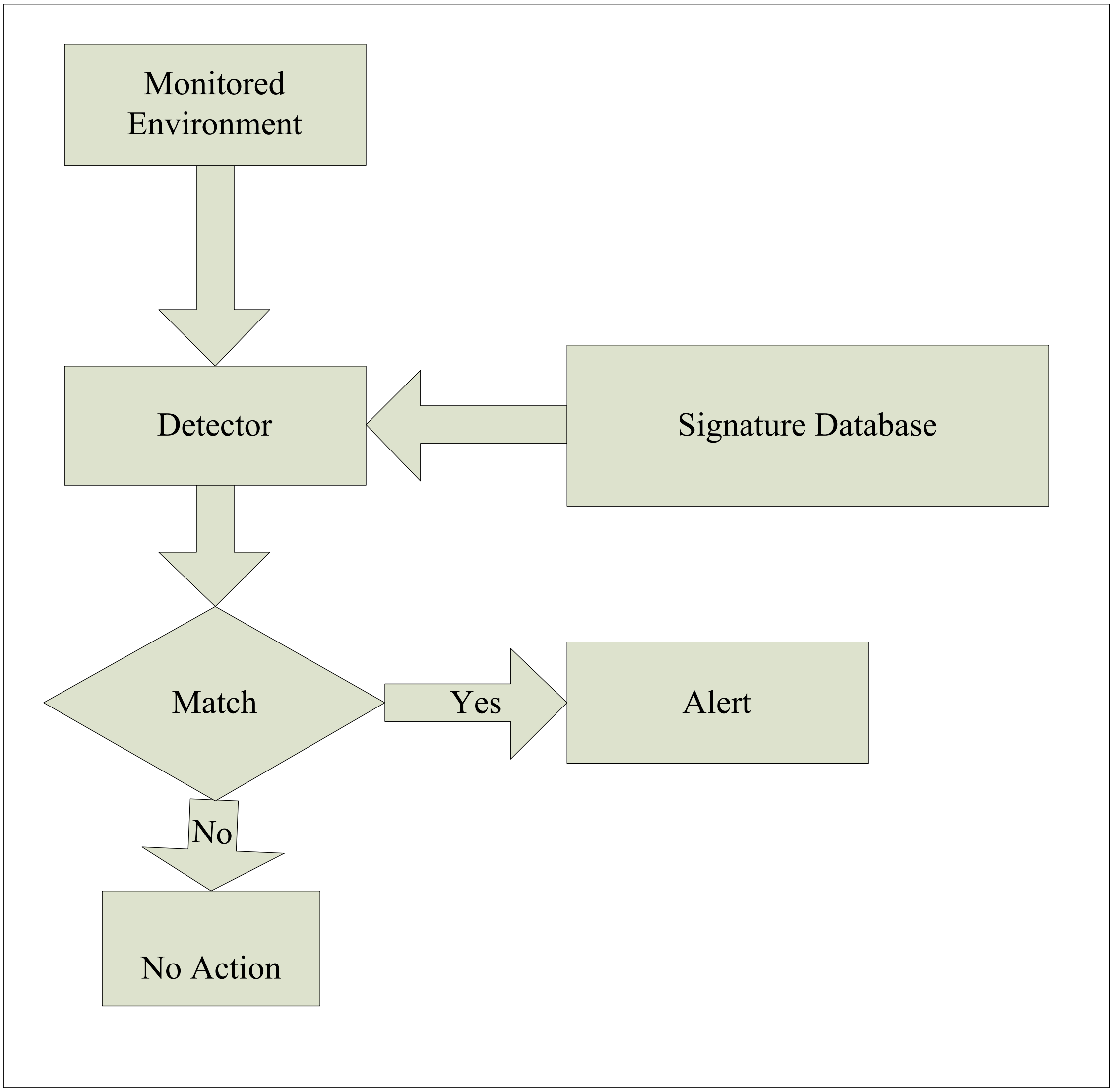

Figure 1.4. Signature based methodology architecture.

Signature based methodology also analyzes the systems calls for known threats payload [7]. Signature based methodology is very effective against know attacks/violations but it cannot detect new attacks until it is updated with new signatures. Signature based IDPS are easy to evade since they are based on known attacks and are depended on new signatures to be applied before they can detect new attacks [15]. Signature based detection systems can be easily bypassed by attackers who modify known attacks and target systems that have not been updated with new signatures that detect the modification. Signature based methodology requires significant resources to keep up with the potential infinite number of modifications to known threats. Signature based methodology is simpler to modify and improve since its performance is mainly based on the signatures or rules deployed [19].

The Signature based methodologies have the following advantages and disadvantages:

Advantages

- Has no learning/training period
- Very efficient at detecting known threats
- Low volume of false positives
- Less overhead on the system being monitored

Disadvantages

- Needs signature update to detect new threats
- Cannot detect variants of know attacks
- Cannot detect insider attacks
- Leaves the monitored environment at risk during the time when a new threat is discovered and the time a signature is applied.

3.2.3 Stateful protocol analysis based methodology. The Stateful protocol analysis methodology works by comparing established profiles of how protocols should behave against the observed behavior. The established protocol profiles are designed and established by vendors. Unlike the signature based methodology which only compares observed behavior against a list, Stateful protocol analysis explores in detail how the protocols and applications should interact/work. This deep understanding/analysis places a very high overhead on the systems [30]. Stateful protocol analysis blends and compliments other IDPS methodologies well which has led to rise of hybrid methodologies [32]. Stateful protocol analysis's deep understanding of how protocol should behave is used as a base for developing IDPS that understand web traffic behavior and are effective at protecting websites [33]. Although the Stateful protocol analysis has a deep understanding of the monitored protocols, it can be easily evaded by attacks that follow and stay within the acceptable behavior of protocols. Stateful protocol analysis methodologies and techniques have slowly been adapted and integrated into other methodologies over the past decade. This has led to the decline of IDPS that utilize just Stateful protocol analysis methodology. The majority of the research on IDPS methodologies mainly concentrates on anomaly, signature, and hybrid methodologies which further reduce the viability of Stateful protocol analysis as a standalone IDPS methodology.

The general architecture of Stateful protocol analysis is shown in Figure 1.5. This architecture is identical to that of the signature based methodology with one exception, instead of the signature database the Stateful protocol analysis has database of acceptable protocol behavior.

The Stateful protocol analysis methodologies have the following advantages and disadvantages:

Advantages

- Has no learning/training period
- Very efficient at detecting known threats
- Low volume of false positives
- Can detect specialized threats

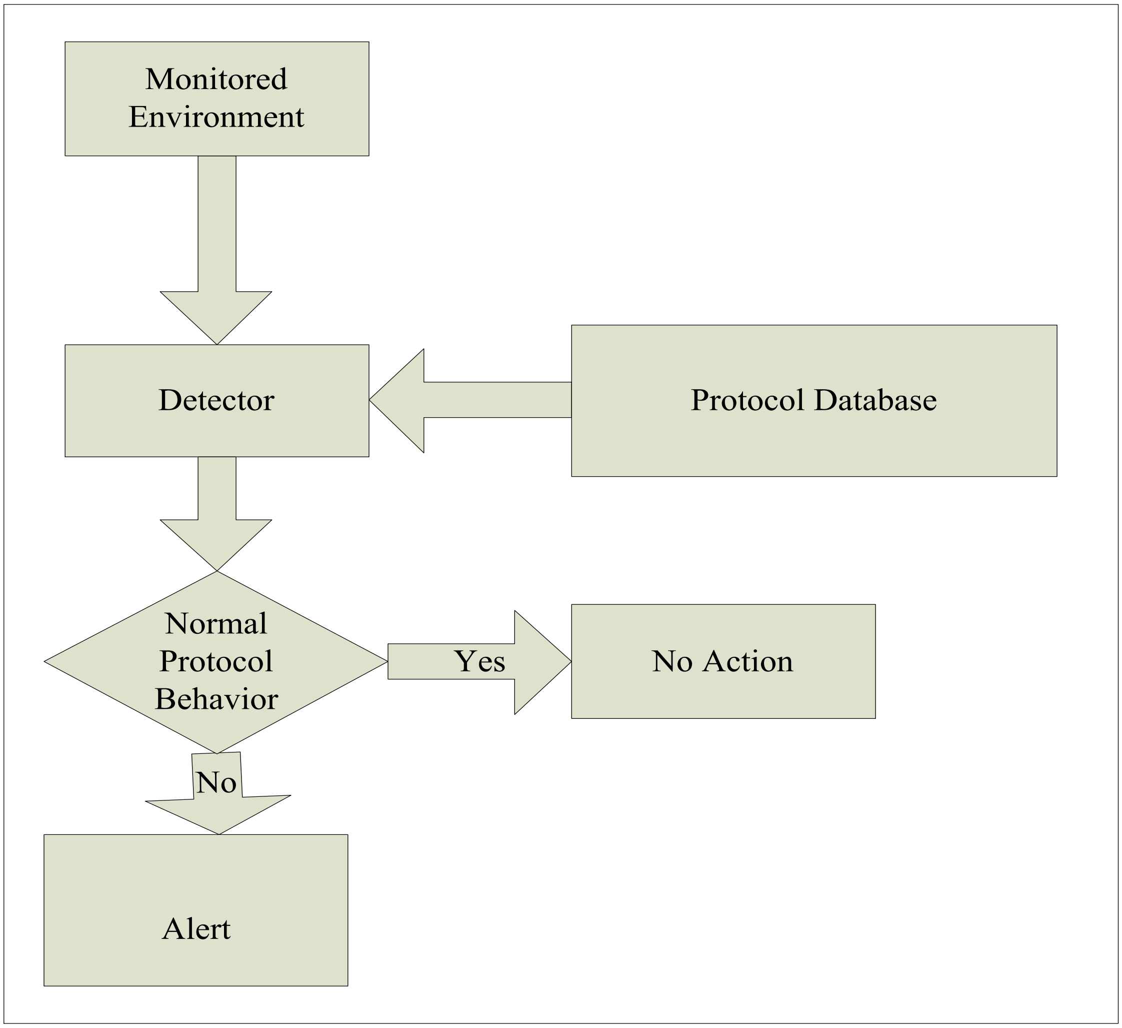

Figure 1.5. Stateful protocol analysis based methodology architecture.

- Resists evasion

Disadvantages

- Needs signature/rule update to detect new threats
- Cannot detect variants of know attacks
- Cannot detect insider attacks
- Can place high overhead on the processing system

- Leaves the monitored environment at risk during the time when a new threat is discovered and the time a signature/rule is applied

3.2.4 Hybrid based methodology. The hybrid based methodology works by combining two or more of the other methodologies. The result is a better methodology that takes advantage of the strengths of the combined methodologies. Prelude is one of the first hybrid IDS that offered a framework based on the Intrusion Detection Message Exchange Format (IDMEF) an IETF standard that allows different sensors to communicate[26]. In [27], Snort is modified by adding an anomaly based engine to its signature based engine to create a better detection and then the new hybrid systems is tested against the regular Snort using same test data. The hybrid system detected more intrusions than the regular one. A hybrid intrusion detection system of cluster-based wireless sensors networks was proposed that worked by breaking the detection into two, first it used anomaly based model to filter the data and then it used signature based model to detect intrusion attempts. Another model for a hybrid methodology was proposed based on how the human immune system works [28]. The proposed system is "based on the framework of the human immune system, that uses a hybrid architecture which applies both anomaly and misuse detection approaches" [7]. A general over view of a hybrid system is shown in Figure 1.6 where three popular methodologies are combined to produce a better system that capitalizes on the strengths of the combined methodologies. The monitored environment is analyzed by the first methodology which sanitizes the monitored environment and then sanitized passed environment is then analyzed by the second methodology which repeats the sanitizing process. After the second sanitizing, the final methodology is engaged to performs the final cleanse. This produces a better system.

The Hybrid based methodologies have the following advantages and disadvantages:

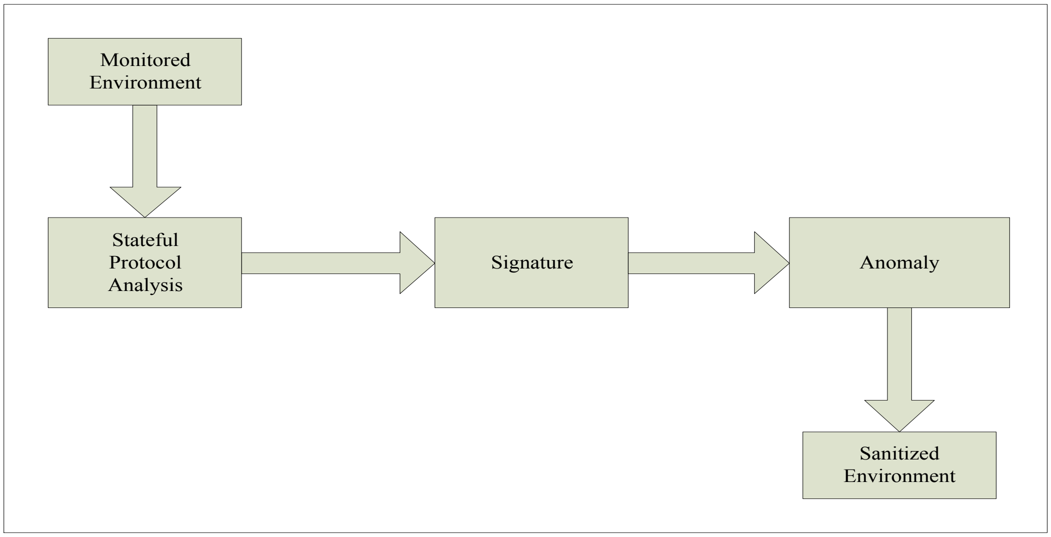

Figure 1.6. Hybrid based methodology architecture.

Advantages

- Has a shorter or no learning/training period
- Very efficient at detecting both known and unknown threats
- Can detect variants of know attacks
- Low volume of false positives
- Accurate alerts
- Can detect insider attacks
- May protect the monitored environment during the time when a new threat is discovered and the time when a signature/rule is applied

Disadvantages

- Needs signature/rule update to detect new threats
- Cannot detect variants of know attacks
- Cannot detect insider attacks

- May leave the monitored environment at risk during the time when a new threat is discovered and the time when a signature/rule is applied.

3.3 Advantages and Disadvantages of IDPS Detection Methodologies Features

The four main methodologies have advantages and disadvantages over one another. The current systems are combining these methodologies in effort to have intrusion detection and prevention system takes advantage of the advantages of each methodology while reducing the short comings of each methodology. This section details the advantages and disadvantages of each methodology. After going through research papers, commercial products, we selected the following parameters to compare the four methodologies. The advantages and drawbacks of the four main intrusion detection and prevention system (IDPS) methodologies are described below and shown in Table 1.1.

3.3.1 Detects new attacks. This is the ability of a methodology to automatically detect new attacks to the protected environment. This should happen without any changes or updates to the monitoring system. A system that uses the anomaly or the hybrid based methodologies can detect new attacks without any updates, while the signature based and the Stateful protocol analysis based systems needs their signatures or rules to be modified and updated with new attacks signatures.

3.3.2. Detects insider attacks. Insider attacks happen when trusted personal who have access and knowledge of the intrusion detection and prevention system takes advantage of their access and knowledge and use it to bypass and attack the system. Only systems that employ the anomaly methodologies can detect the diversion from the normal behavior of a user and alert on it. Other methodologies cannot detect this change in behavior as they are only looking for known signatures or behavior.

3.3.3 Detects attacks from day one of installation. This is the ability of an IDPS to work from the moment of initial installation without needing a period of time for the methodology to learn the environment and create profiles. The signature and the Stateful protocol analysis based methodologies have the advantage here when compared to the anomaly based methodology. The hybrid based methodology can have an advantage depending on the characteristics of the combined methodologies.

3.3.4 Detects all known attacks. This is the ability of the methodology to successfully detect all know threat attempts. This is an area were the signature and the Stateful protocol analysis based IDPS has an advantage over the anomaly based systems. The hybrid based methodology can have an advantage or disadvantage based on the methodologies that are combined.

3.3.5 Detects variants of known attacks. This is the capacity of the methodology to recognize any modifications and variants of all known attacks. The anomaly based methodologies have the advantage over other methodologies due to the way the anomaly works. Signature and Stateful protocol analysis based methodologies cannot detect any changes to know attacks without first updating their signatures or rules. The hybrid based methodologies can have an advantage if one of their combined methodologies is anomaly based.

3.3.6 Training period. A training period is the amount of time required by an IDPS methodology after installation to learn the monitored environment and build its profiles. These profiles are then used by the methodology as a base. This is a disadvantage for the anomaly based methodology as it prevents it from working from day one of installation. This can also be a disadvantage for the hybrid methodology based if anomaly based methodology is one of the combined methodologies, but the disadvantage is not as high as for the anomaly based methodology.

3.3.7 Easy to understand alerts. An IDPS methodology has to produce threat alerts that are easy to understand. The anomaly based methodology is plagued with a very high number of alerts due to the way it works and this makes it difficult to interpret and understand all the alerts it produces. The signature based and Stateful protocol analysis based methodologies tend to generate few and more specific alerts which gives them an advantage over the anomaly based and hybrid based methodologies.

Table 1.1

Advantages and disadvantages of intrusion detection and prevention system (IDPS) methodologies

Advantage(Yes)/Disadvantage(No)	**Hybrid**	**Anomaly**	**Signature**	**Stateful Protocol Analysis**
Detects new attacks	Yes	Yes	No	No
Detects insider attacks	Yes	Yes	No	No
Detects attacks from day one of installation	Yes	No	Yes	Yes
Detects all known attacks	Yes	Yes	Yes	Yes
Detects variants of known attacks	Yes	Yes	No	No
Training Period	No	Yes	No	No
Easy to understand alerts	Yes	No	Yes	Yes
Needs signature or rule updates	Yes	No	Yes	Yes

Depending on which methodologies a hybrid methodology combines, a hybrid system can produce better alerts than the single methodologies.

3.3.8 Needs signature/rule updates. This is a measure of how an IDPS methodology stays current. It requires constant signature or rule updates or it continuously update itself. The anomaly based methodologies has the advantage over others as it monitors the environment and updates its profiles by itself. While the signature and the Stateful protocol analysis based methodologies require constant signature or rule updates. The hybrid based methodologies have a slight advantage over the signature and Stateful protocol analysis based methodologies. The hybrid advantage is depended on the combined methodologies.

3.4 Evaluations of IDPS Detection Methodologies

This section offers a description of ways for evaluating intrusion detection and prevention system (IDPS) methodologies and the systems that are based on these methodologies. Table 1.2 can be used to evaluate any intrusion detection and prevention system (IDPS) whether it uses one of the three main methodologies or a combination of the two or more of the other methodologies.

3.4.1 Resistance to evasion. The intrusion detection and prevention system (IDPS) should be able to detect evasion attempts and stop them. These attempts are more common with the signature and stateful protocol analysis based intrusion detection and prevention system (IDPS) due their dependence on signatures. Anomaly based intrusion detection and prevention system (IDPS) have better resistance to evasion, but the hybrid based system offers the best resistance to evasion attempts due to the combination of other methodologies.

3.4.2 High accuracy rate. An IDPS should have a high accuracy rate when detecting and analyzing possible threats. The signature based methodology has a high accuracy rate on known threats but its overall rate is lower than the anomaly based methodology which can detect threats which can detect previously unknown threats. The hybrid based methodology offers the best accuracy rates.

3.4.3 Market share. Market share is the measure of the methodology's dominance in the deployed systems. The signature based methodology far outweighs the other three methodologies, followed by Stateful protocol analysis. The anomaly and hybrid based methodology are the bottom but their adaption is growing much faster and will soon surpass the first two methodologies.

3.4.4 Scalability. Scalability is the ability of an IDPS to scale and grow with environment once deployed. The signature and Stateful protocol analysis based methodologies are easy to scale since they are based on signatures that can be easily scaled. A hybrid based methodology can be easily scale depending on the underlying methodologies. The anomaly based methodology is the least scalable methodology due the time it requires to learn and build its baseline profiles.

3.4.5 Maturity level. Maturity level looks at how long a methodology has been around and how stable it is. The signature based methodology is the most mature, followed by the Stateful protocol analysis and anomaly based methodologies. The hybrid methodology is at the bottom of this list, but it is growing at a much faster than the others.

3.4.6 Overhead on monitored system. The intrusion detection and prevention system (IDPS) should not place a lot of overhead on the monitored systems; it should work without affecting the performance of monitored systems. Signature and Stateful protocol analysis places the least overhead on the monitored systems. The hybrid based methodology can place a high overhead burden on the monitored system depending on the combined methodologies. The anomaly based methodology places the most overhead on the monitored system.

3.4.7 Maintenance. The anomaly based methodology requires the least amount of maintenance since it does not require updates to detect new threats. The other three methodologies require constant signature updates in order to keep up with new threats. This constant updating of signatures adds to the resources required to maintain the methodology.

3.4.8 Performance. The intrusion detection and prevention system should be able to perform at peak performance under all condition on the monitored system without becoming a bottle neck or reducing its efficiency. The signature and Stateful protocol analysis based methodologies offers better performance than anomaly and hybrid based methodologies since they only check for well-defined signatures which do not require as much resources.

3.4.9 Easy to configure. The intrusion detection and prevention system (IDPS) should be easy to install and integrate with other security tools already in the environment. The signature and the Stateful protocol analysis methodologies are easier to install and configure. They do not require as much time to tune since they use signatures that can be updated automatically in some cases. The anomaly and the hybrid depending on the combined methodologies require more time to configure, learn, and tune the environment.

3.4.10 Easy to use. The intrusion detection and prevention system should be easy to use and understand. This means it produces less false positives and false negatives which makes it easier to analyze and understand the alerts. The signature and the Stateful protocol analysis methodologies are easier to use since they produce fewer alerts. The hybrid based methodology can be easier than the anomaly depending on its underlying methodologies. The anomaly requires more resources to manage the high volumes of alerts it produces.

3.4.11 Protection against new attacks. The intrusion detection and prevention system should be able to detect new threats. The anomaly based methodology does detect new attacks

Table 1.2

Parameters for evaluating IDPS methodologies.

	Anomaly	**Signature**	**Stateful Protocol Analysis**	**Hybrid**
Resistance to Evasion	Medium	Low	Low	High
High accuracy rate	Medium	Medium	Medium	High
Market Share	Medium	High	Medium	Medium
Scalability	Medium	High	High	Medium
Maturity Level	High	High	High	Medium
Overhead on Monitored System	Medium	Low	Low	Medium
Maintenance	Low	Medium	Medium	Medium
Performance	Medium	High	High	Medium
Easy to Configure	No	Yes	Yes	No
Easy to Use	Medium	Low	Low	Low
Protection against New Attacks	High	Low	Medium	High
False Positives	High	Low	Low	Low
False Negatives	High	Medium	Medium	Low

without any updates unlike the signature and Stateful protocol analysis that require their signatures to be updated before they can detect previously unknown threats. The hybrid based methodology can detect new threats if one of the underlying methodologies is anomaly based.

3.4.12 False positives. False positives happen as a result of a methodology misclassifying a non-threat event as a threat. The anomaly based methodology is plagued by false positives. The signature and Stateful protocol analysis based methodologies produces the least number of false positives. The hybrid based methodology's level of false positives is low if anomaly based is not part of its underlying methodologies.

3.4.13 False negatives. False negatives are a result on a methodology classifying threats as non-threats. The anomaly based methodology produces the most false negatives when compared with signature and the Stateful protocol analysis based methodologies. The hybrid based methodology produces less false negatives if it does not use anomaly based methodology as one of its underlying methodologies.

CHAPTER 4

Setting Up IDPS Testing Environment Using Tomahawk and Wireshark

4.1 Introduction to Tomahawk and Wireshark

Tomahawk and Wireshark are open source tools that we used to create and manipulate the files during our evaluations.

4.1.1 Tomahawk. Tomahawk is an open source network tool that can be used to generate background network traffic, replay network traffic, and manipulate network traffic using captured network traffic files. The captured traffic can then be replayed during the evaluation of an IDPS. For the evaluation to be effective the traffic capture should come from the environment where the IDPS will reside. This produces a more accurate test. Once the traffic is captured, it can be replayed in controlled environment where more experimentation can be done without negatively affecting the production environment. Before the capturing is started care should be taken to guarantee that all session information was also capture [41]. A simply way to make sure that all session related information is captured is to start Wireshark and the recording before the attacks are launched. Within the controlled environment the captured traffic can be used as background traffic while exploits are introduced to the monitored workstation/server.

4.1.2 Wireshark. Wireshark is open source network utility that available at no cost form it web site. It is mainly used for analyzing network packets and network troubleshooting [43]. Wireshark has other uses such as capturing network traffic and saving in a number of formats. We selected it Wireshark for use in capturing and saving attacks in our test environment. Among the reasons we selected Wireshark is that it has a very small foot print and it runs on Windows and has nice graphical user interface when compared to other utilities that can capture and save network traffic. Wireshark is also very stable and has big user base and it is well document and for our purposes we just click the start and stop recording buttons and then save the output in a format we want.

The following are the hardware and software requirements for using Tomahawk to evaluate an IDPS.

4.2 Hardware and Software Requirements

- Workstation or server with a minimum of two network cards (running Linux/Unix flavor operating system)
- Second workstation or server with a minimum of two network cards
- Captured network traffic file in the libpcap format (cleaned)
- Network traffic capture tool (Wireshark)
- A minimum of a 2.0 GHz Pentium equivalent processor
- A minimum of 1GB of ram (2 or more is recommended)
- Network switch (optional)
- Management pc (optional)
- Third network card on the other workstation/server (optional)
- Network cables (crossover optional)

4.3 Hardware Setup

There are three ways of configuring the hardware for testing an IPDS using Tomahawk. A basic way with just two computers with a minimum of two network cards each, a medium setup which is a basic setup with a hardware IDPS, and an advanced way which is a medium setup with an addition of a network switch and a management computer. These setups do not include an internet connection as a way to control the network traffic during the evaluation of an

IDPS using tomahawk. If desired an internet connection can be added to either the attacker or the attacked computer.

4.4 Basic Setup

A basic setup is the simple way to test a software based IDPS. As shown in Figure 2.1, only two machines with two network cards and two crossed-over network cables are required. One of the machines is configured as the attack machine and Tomahawk is installed on it. The other machine is configured as the attacked machine and the software based IDPS is installed on it. The two machines are connected to one another with the crossover cables. Tomahawk only runs on Unix or Linux based operating systems and as a result the attack machine will require a Linux or Unix based operation system. The attacked machine can run any operating system.

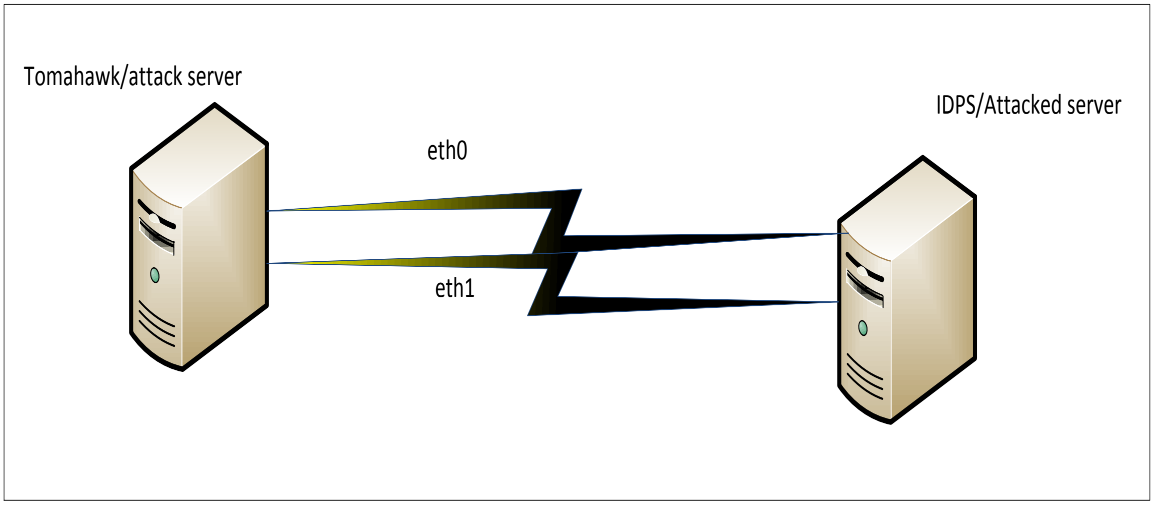

Figure 2.1. Basic Setup

4.5 Medium Setup

The medium setup is also simple but adds a hardware/appliance based IDPS. It requires two machines with two network cards each and three network cables. The machines are connected through the IDPS as shown in Figure 2.2. In this setup the IPDS saves as a network switch connecting the two computers. The computer that tomahawk will be installed on has to have a Linux or Unix based operating systems. Also the computer that has tomahawk running on it must have two network cards that will be used by tomahawk. The other computer that serves as the attacked one can have any of the current operating systems.

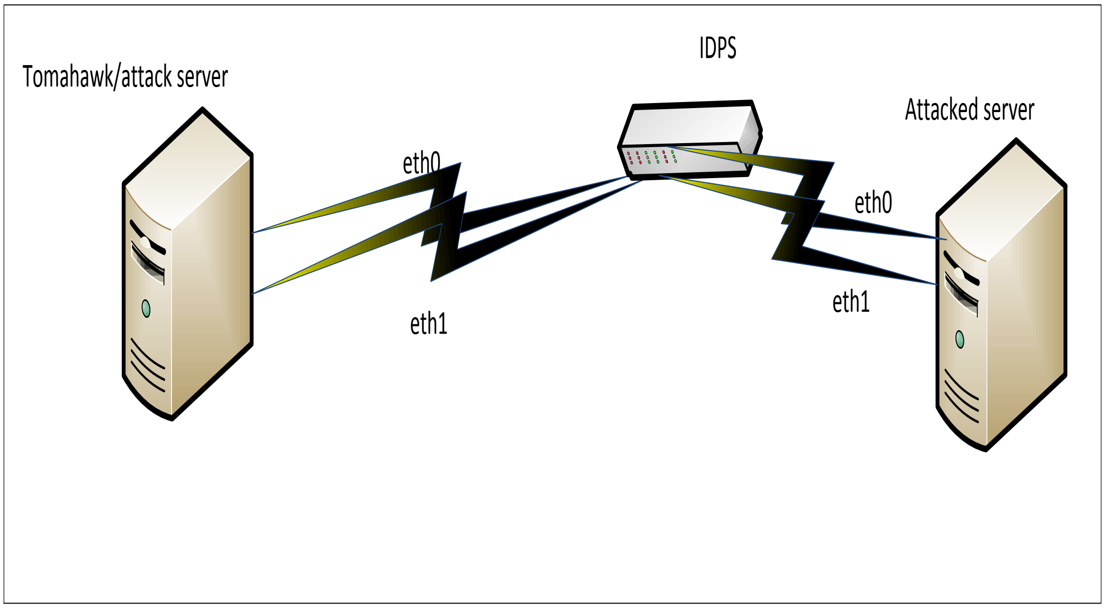

Figure 2.2. Medium Setup.

4.6 Advanced Setup

The advanced setup is more involved than the other two as it adds a third network card, a switch, five network cables, a management machine, and the internet. The machines are connected through the switch and the IDPS as shown in Figure 2.3. One of the machines with three network cards is configured as the attack machine and Tomahawk is installed on it. The other machine is configured as the attacked machine and sets on behind the IDPS and a third management machine is connected to the switch. This the ideal setup for testing IDPSs as it allows for different configuration changes to be made. For example, more computers can be added to the test by adding another switch between the IDPS and the attacked computers or by adding more computers on both sides. This would allow for evaluating the IDPS behavior under high network traffic.

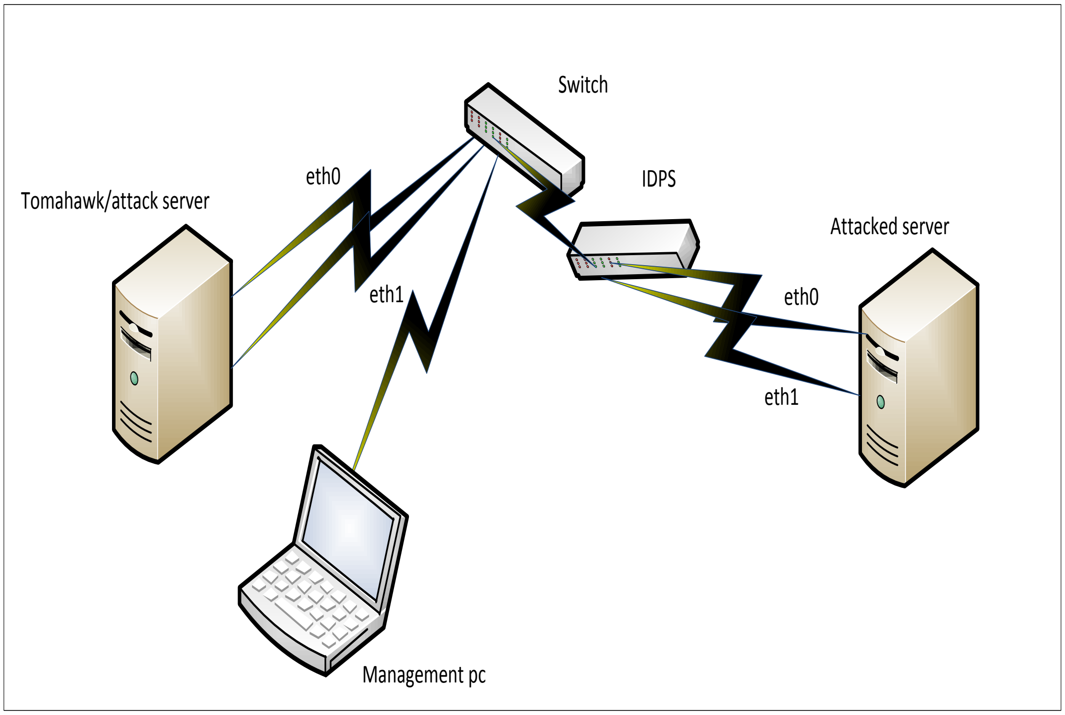

Figure 2.3. Advanced Setup.

4.7 Advantages of Using Tomahawk

Tomahawk was chosen for this setup due to the advantages it offers over other tools that replays captured network packets. Tomahawk uses simple commands and flags that can be teamed together to easily manipulate the traffic going to the attacked computer. It can take a small packet and manipulate it to produce the desired traffic flow.

Some advantages of using tomahawk include:

- Tomahawk is free and publicly available
- It is simple to use
- It is very stable and mature
- Does not require a lot of resources to run it
- Can evaluate both software and hardware based IDPS

4.8 PCAP File

A PCAP file is a file that contains captured network activity and saved in the *libpcap* format with a PCAP extension. This format and extension allow the file to be used by multiple network related tools on most current operating systems. Tomahawk works by replaying and manipulating PCAP files. Tomahawk does not create its own PCAP files but they can be created by network monitoring tools such as Wireshark. Wireshark is an open source network monitoring tool that has multiple functions and it runs on most current operating systems. PCAP files can also be downloaded from the Internet from trusted sources. There are advantages to creating your own PCAP file for use with Tomahawk. Using own PCAP files allows to use traffic that is representative of the environment where the IDPS will reside and protect. It also allows the capture of traffic at different times and different load situations. This facilitates different mixes of traffic volumes and applications on the network. Using a downloaded PCAP may not present a true picture of the environment being tested which can led to a wrong IDPS been chosen.

4.9 Testing Methodology

Tomahawk can be configured and used in variety ways to support different test configurations. The three setups described above are examples of setting up three different test environments for evaluating an IDPS using Tomahawk. Tomahawk works the same way regardless of the configuration of the setup and it supports both software and hardware based IDPS. Tomahawk works by replaying captured network packets that are saved as a PCAP file in a bi-direction fashion and breaking the PCAP file into two pieces and then assigns these pieces to the client and the other to the server [44]. Using this system allows Tomahawk to keep track of the PCAP file as it is replayed. By breaking the PCAP file into packets allows Tomahawk to assign the first IP address it encounters in the PCAP file to the client and the second IP address to the server. This process is repeated until the whole PCAP file is replayed entirely. Once the PCAP file is broken down into packets and the client and server IP addresses are assigned, tomahawk starts replaying the packets. The client packets are sent out on eth0 and the server packets are send out on eth1. Tomahawk has a default of 0.2 seconds for re-transmissions of lost packets and it also auto manages other network related tasks such as MAC addresses, client and server IP address. If the IDPS detects and blocks the PCAP file that contains an attack, tomahawk will report a time out. If tomahawk reports that the PCAP containing an attack completed without any errors, then the IDPS will have missed the attack [44].

4.10 Creating PCAP Files

We use Wireshark to capture and create four PCAP files that we use in our test environment. Creating PCAP files with Wireshark is documented in [43]. The PCAP files we use were created using default settings in Wireshark. After starting Wireshark, we recorded the network traffic and then initiated the attacks/exploits that way we capture all the packets related to the attack. The following PCAP files were created:

4.10.1 PCAP1. This is a simple file that contained normal network traffic and no attack traffic. This file is a capture of traffic browsing a server. This capture will be used to test how the IDPS handles normal traffic and establish some baselines.

4.10.2 PCAP2. This file is a capture of a known OS exploit and will be used to test if the IPDS will detect and respond to the attack.

4.10.3 PCAP3. This file contains a DOS attack on the server and will be used to test how the IDPS detects and responds to the attack.

4.10.4 PCAP4. This file is a capture of an exploit and a DOS attack while there is high volume of traffic on the network. This file will be used to verify how the IDPS reacts under different situations.

4.11 Using Tomahawk

Tomahawk is a command line utility that runs on Linux based operating systems. Tomahawk commands can be used to run a basic evaluation on an IDPS. To use Tomahawk just type tomahawk on the command prompt followed by any of the flags. A detailed explanation of Tomahawk's every command and flag is detailed in [44]. In our test setup, we used the Medium setup described above with the following hardware and software:

- The attack workstation- IBM Workstation running SUSE Linux
- A switch that has DHCP and IDPS capabilities
- The attacked server- An IBM Workstation running SUSE Linux
- Four network cables
- The attack workstation and the attacked server were connected through the switch/IDPS. Care was taken to make sure that all the traffic from the attack workstation to the attacked server passed through the IDPS.

The first test involved the PCAP1 been replayed against the attacked server. The following Tomahawk commands were used for testing:

```
tomahawk -l 2 -f pcap1.pcap
```

This command replayed the pcap1 file twice and produced the following output:

```
Beginning test

Completed 1 loop of trace pcap1.pcap

Completed 1 loop of trace pcap1.pcap
```

```
Finished 2 loops of trace pcap1.pcap Completed: 2, Timed out: 0 Retrans: 0 Sent: 1686 Recv: 1686
```

This output shows that the both replays finished without being blocked. If the pcap1 file was blocked/dropped by the IDPS then the loop will have not completed.

```
tomahawk –l 2 –f pcap2.pcap
```

The above command replayed PCAP2 file which contained a known exploit against the server. The IDPS blocked this attack and the packets were dropped. As a result the loops did not complete.

```
tomahawk –l 2 –f pcap3.pcap
```

The above command replayed PCAP3 file which contained a DOS attack against the server. The IDPS blocked this attack and the packets were dropped. As a result the loops did not complete.

```
tomahawk –l 2 –f pcap4.pcap
```

The above command replayed PCAP4 file which contained an exploit and a DOS attack while there is high volume of traffic on the network. These replay packets were dropped by the IDPS and as result the loops did not complete.

CHAPTER 5

Experimental Analysis of IDPS Products

5.1 Selecting IDPS Products:

For the test environment we selected four IDPS products; two that are open source and software based and two that are proprietary and appliance based. After researching Internet, we found the plethora of IDPS products; both open source, as well as commercial. We provide a partial list of these products in table 2.1. This table is current as on 05/21/2012. This list is dynamic due to products being bought by other companies and in the case open source products, customer support becomes unavailable or the product does not keep up with new developments and competing products which lead users to abandon the product and flock to newer products or products with better support. The main driving factors for selecting these products were maturity and availability. The section below lists other reasons for choosing these IDPS products. These four products were also evaluated based on the parameters presented in table 1 in chapter 2.

5.1.1 Snort (IDPS 1). Snort was chosen due to its maturity, scalability, cost, and market share. Snort is an open source based IDPS product that is available to the public free of charge and can be downloaded it from Snort's website Snort [48]. Snort has been around for more than fourteen years and it is the leading IDPS and is used as a base for a number of other open source and commercial IDPS [48]. Snort has a large community of users that offer support through forums and it also gets development and upkeep support from Sourcefire. Sourcefire is a company that sells a commercial version of Snort and provides commercial support for a fee [49]. Snort is easy to install and it is not resource intensive and can be installed on any current computer system that has a Linux based operating system. Snort can also be installed on a Windows based operating system [50]. We also selected Snort due to its support of multiple detection methodologies and its ability to read and examine a PCAP file as if it was monitoring the packets off the network [48]. This offers other evaluating opportunities that are not available in other IPDS products.

5.1.2 OSSEC (IDPS 2). OSSEC is another open source based IDPS product that we selected and used in our test environment. OSSEC was select based on how it works, it is mainly an anomaly based product that works by checking log analysis, inspecting and maintaining file integrity, policy monitoring, and rootkit detection [51]. This allows OSSEC to offer a more accurate product although it can produce a high volume of alerts depending on how it is tuned. OSSEC supports all of the current operating systems and it has a very small foot print which allows it to run on any hardware. It is simple to install and configure and is highly scalable by design. OSSEC can run as a standalone IDPS protecting a single machine or as centralized protecting multiple machines with different operating systems [51]. OSSEC was acquired by Trend Micro a commercial maker of computer security products and have promised to continue the OSSEC project but added commercial support and development to it. OSSEC was also selected for other possible uses in future work due to its varied uses.

5.1.3 Proprietary 1 (IDPS 3). IDPS 3 is one of the commercial proprietary IDPS products that we had access to use in our test environment on condition we do not disclose its identity during our evaluations. It is for this reason that we decided to refer to all the IDPS products in our environment as IDPS and a number. The first proprietary product, IDPS 3 is an appliance based product that is very easy to install and use. It is a plug and play IDPS product. IDPS 3 offer a management console accessed through a web browser that offers advanced configuration and customizations. Although the console is simple to use, an advanced networking security skill is required in order to correctly customize the product. One of the drawbacks of this product was that it was an entry-level product.

5.1.4 Proprietary 2 (IDPS 4). IDPS 4 is the second commercial proprietary IDPS products that we had access to use in our test environment on condition we do not disclose its identity during our evaluations. Unlike IPDS 3, IDPS 4 is a high end commercial IDPS that uses hybrid detection methodologies and is only available as an appliance. IDPS 4 comes with paid support that   allows you to get updates and signatures. This is a very expensive product to purchase and support. IDPS 4 comes with help on initial installation, configuration and customization. IDPS 4 can be easy to use since it can be configured to automatically receive signatures and has a roll back feature. IDPS 4 has a number of options that come at additional cost. The major option is the ability to inspect encrypted packets but this places significant overhead on the product.

5.2 Setting up Testing Environment

We conducted all our tests using the advanced setup, described in section-4. The other setups outlined in section 4 were not adequate for our needs and that is why they were not chosen. The advanced setup in section 4 was used as a guide for creating a more involved advanced setup that is representative of a production environment with diverse hardware and software as shown in Figure 3.1 and Table 2.2. In our test environment setup, we expanded the advanced setup so that we will be able to try out different configurations without making major configuration changes. For example, we could send identical attacks to four workstations that are behind different IDPS products without making configuration changes. This is possible due to the high number of available workstations and it allows us to also observe and compare the performance of the workstations, network, and IPDS products while under attack. We stayed away from using virtual machines as they complicate performance calculations due to the sharing of resources. It could be easy to confuse overhead from the attacks and overhead created by the host as it tries to manage the demand by the virtual machines as they become under attack

Table 2.1

Sample IDPS products

Product Name	**Product Type** **Software/Appliance**
Snort	S
OSSEC	S
CheckPoint	Both
Proventia	A
Tipping Point	A
Cisco IPS	A
Symantec Endpoint Protection	Both
Sourcefire NGIPS	S
AirDefense	Both
Airtight	S
NitroGaurd IPS	A
StoneGate IPS	Both
Bro	S

Table 2.2

Hardware, software, and utilities used in our environment

Hardware	**Operating Systems**	**Networking Utilities**	**Hacking Utilities**
4 IBM Workstations	SUSE Linux	Wireshark	Backtrack
3 Dell Servers	Windows 2008 R2	Tomahawk	LOIC
Cisco Router	Windows 7	WhatsUPGold (evaluation copy)	Jolt2
2 Cisco 48 port switches		Internet Access	
4 ThinkPad laptops		IDPS 1	
2 Dell Latitude laptops		IDPS 2	
IDPS 3			
IDPS 4			

Using the above hardware and software we built a test environment shown in Figure 3.1. The internet, firewall and the router where only used during the setup phase of the test lab. During this phase only a Windows 7 laptop was connected and this laptop was used to download all the required software. Once all the required software was downloaded the router and the Windows 7 laptop were disconnected from the switch. This was done as a way to control the network traffic that will be present on the test environment network during our evaluations. Running the experiments while connected to the internet could produce unreliable results since the traffic patterns would vary between experiments. As a result we setup our lab, performed, and captured the attacks to be used with the experiments without the internet. This assures that our attacks do not contain other traffic or attacks from the internet that we are not aware of. The Internet connection can be connected back during the test phase so that it can be used for background traffic and producing a realistic test environment if the production environment where the IDPS will reside is connected to the internet.

Our test lab separated the network switches and placed the IDPS between the switches. Switch 1 provides connectivity to the internet, management laptop, attack machines, and the server that hosts Tomahawk. On the other hand we have Switch 2 and it provides connectivity to all the machines on the attacked side. Between Switch 1 and Switch 2 are the IDPS products that link the two switches. We placed the IDPS between the two switches so that all the traffic between the attack and attacked machines go through the IDPS.

We evaluated four different IDPS products, two software based and two that are hardware based. These products have been described in the preceding section.

5.2.1 Hardware setup. The hardware, software, and tools used in our test lab are listed above in Table 2.2 and preceding section went over the general setup of out lab. In this section we describe Figure 3.1 which is how the hardware in the test environment is laid out. The IDPS products divide the network and only one IDPS is shown in Figure 3.1 for simplicity reasons. The lab has four IDPS products, three that are placed were IDPS is shown in Figure 3.1 and one that is place on the attacked workstation.

5.2.2 Tools. We separated and grouped the tools used in our test environment as shown in Table 2.2 and a brief discussion of some of them is given below.

5.2.2.1 WhatsUPGold. WhatsUPGold is commercial available monitoring tools that we used to monitor the performance of the computers and the network [52]. This tool was used to create base lines and these baselines were used to evaluate performance during the analysis phase. We used an evaluation edition of this product.

5.2.2.2 Backtrack 5. Backtrack is an all in one network security tool that is mainly used to test how secure networks are [53]. We used the tool to launch attacks that we then captured using Wireshark and saved as PCAP files. These PCAP files were then used with Tomahawk to evaluate the IDPS products in our test environment.

5.2.2.3 LOIC and Jolt2. These are related tools that we used to launch denial of service attacks. LOIC launches the standard DOS attack while Jolt2 launches a DOS attack that uses fragmentation [54].

5.2.3 Attacks. Four attacks were used in our evaluations and these attacks are more involved than the attacks covered in chapter 3. We used the following attacks:

5.2.3.1 Denial of service (Attack 1). It is a basic denial of service attack against a Windows server.

5.2.3.2 Complex attacks (Attack 2). In complex attacks we used a combination of attacks and used a distributed denial of service attack to other machines on the network and also reduced the network throughput to speed up the effects of the attacks.

5.2.3.3 Fragmentation (Attack 3). Fragmentation is an attack that takes advantage of the normal network process which allows packets to be broken up and transmitted in bits that are out of order.

5.2.3.4 Evasion (Attack 4). In this attack, an encrypted payload was pushed and delivered to the protected server and once there it was used to attack other servers and workstations on the network.

5.3 Experimental Results

This section describes the results of analyzing the four IDPS products based on the parameters described in Table 1.2 in section 2.

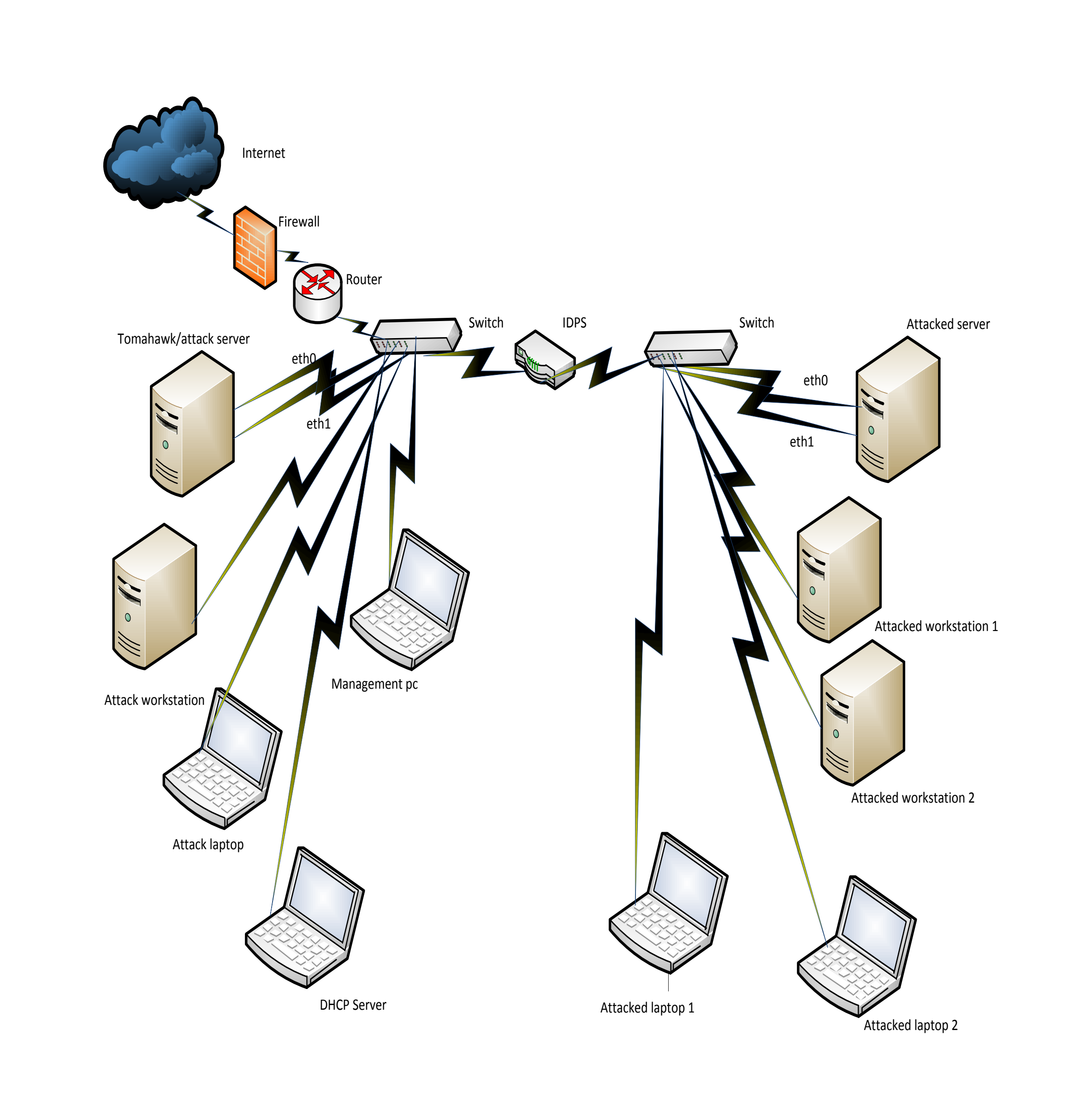

Figure 3.1. Our Test Environment.

5.3.1 Evaluating resistance to evasion. There are a number of ways to evade an IDPS and this number changes as new ways and solutions to old ones are discovered. We choose to evaluate the four IDPS products we have in our test environment using four of the most common techniques. These techniques are denial of service, complex attacks, fragmentation, and evasion.

During the denial of service attack the hardware based IDPS did better than the software ones. Denial of Service is an attack that is designed to cause the targeted system to waste its resources on a diversion caused by the attack thereby eliminating available resources to legitimate system uses/users. Software based IDPS products require an operating system to host them. Depending on the operating system used, an attacker can easy attack the host operating system and disable the IDPS. These are attacks can be as simple as getting the host machine to process other tasks that are resource intensive which limits the resources that are available to the IDPS. In our test we launched Windows 7 exploit using Backtrack and we were able to overwhelm the host that hosed IDPS 2. Once the host was starved of resources, the IDPS failed to stop the subsequent attacks. The appliance based IDPS has a much smaller attack surface since their operating system a custom configured for one task and have all unnecessary services disabled. This gives the appliance based IDPS an advantage over a software IDPS and if the host operating system is Windows based, it fairly easy for the attacker to disable it. IDPS 1 is a little secure compared to IDPS 2 due to its host operating system. UNIX based operating systems are more secure that Windows ones. When we configured IDPS 1 on a Windows host, we were able to get more successful attacks through by first attacking the host. We were able to attack the Unix based host but those attacks required more resources than the Windows ones. The windows hosted IDPS 2 was the only one the managed to spot the evasion attack we launched. In this attack, an encrypted payload was successfully delivered to the protected server and once there it was used to attack other servers and workstations behind the IDPS. IDPS 1, IDPS3, and IDPS4 did not stop the evasion attack since it was encrypted. IDPS 2 allowed the attack to go through but alerted when the encrypted payload was unencrypted and tried to compromise the host. IDPS3 and IDPS 4 have optional modules that can detect and inspect encrypted traffic, but these come at a significant cost. The modules also affect performance as the IDPS spends more resources decrypting and re-encrypting the network traffic as it travels through it.

In complex attacks we used a combination of attacks and used a distributed denial of service attack to other machines on the network. We reduced the network speed to 10MBS. Under these conditions it did not take long for systems to fail. This was designed to place a more work load on the IDPS. IDPS 1 did not perform well under these attacks when compared to the other three. Once the network was pushed the limit, IDPS 1 became unreliable due to the rate of dropped packets. IDPS 2 performed better than the rest of them in this evaluation, but its results need interpretation. IDPS 2 let all the attacks through the machines it was protecting but it stopped the attacks from executing. Fragmentation is an attack that takes advantage of the normal network process which allows packets to be broken up and transmitted in bits that are out of order. An attacker can take advantage of this by sending packets that are out of order. This also places an overhead on the IDPS as it now has to spend more resources processing these packets. IDPS 2 performed poor on this attack and it allowed all of these to go through. The results of the four evaluations are shown in Figure 3.2.

5.3.2 High accuracy rate. We measured accuracy by counting the number of attacks an IDPS successfully detected and blocked. IDPS 3 had the highest number of stopped attacks while IDPS 2 and IDPS 4 were tied in the middle with IDPS 1 at the bottom of the list with the lowest number of foiled attacks. IDPS 2's accuracy rate was misleading and needed to be further analyzed.

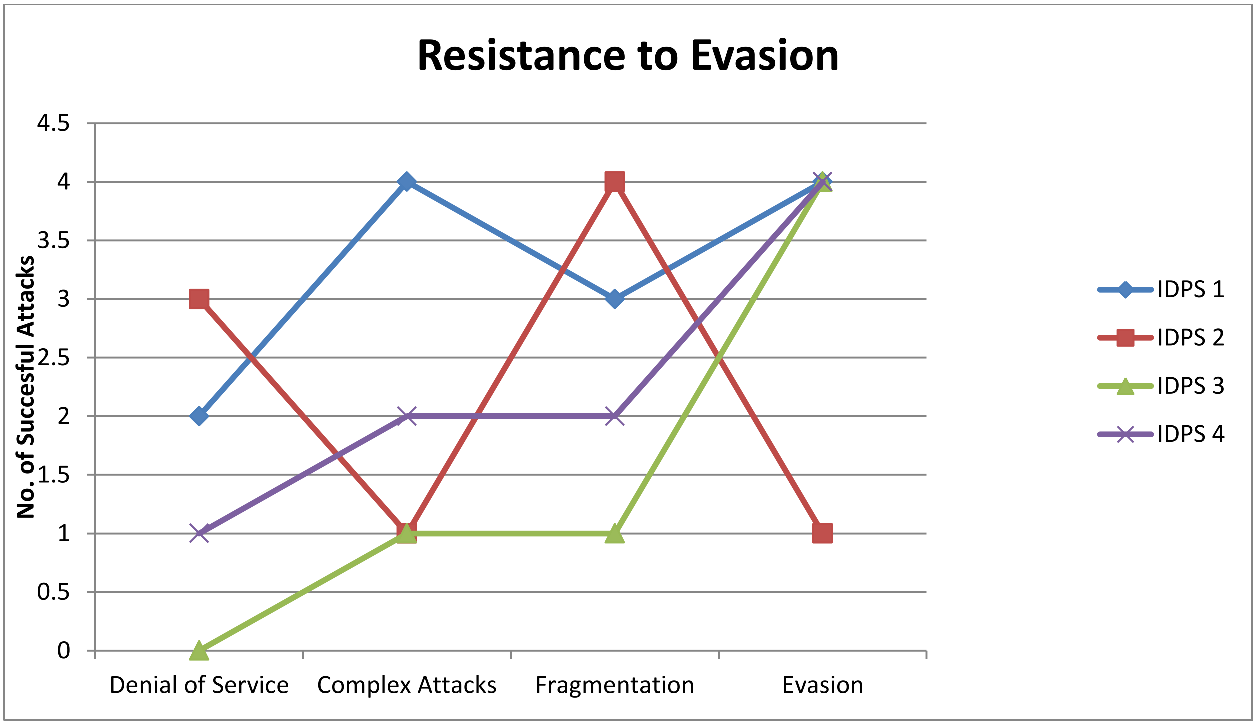

Figure 3.2. Resistance to evasion.

IDPS 2 allowed attacks to get to the protected host but it alerted on all attacks. This is due to the way it works, it only alerts after the attack has reached the host it is protecting and then tries to blocks the attack. This puts it at a disadvantage when compared to an IDPS that detects and blocks attacks on the wire before it gets to the protected host. Once the attack/attacker is on the target machine is much harder to block it. The attack/attacker can easily disable the IDPS or modify the alerts and logs and in so doing cover their tracks, once they are the target machine. The target can also be used as a launching pad for other attacks. IDPS 2 has a very high and accurate detection rate over the other IDPS and it can be a best fit in an environment where all changes to the hosts have to be recorded. Figure 3.3 shows how the four IDPS performed with our attacks.

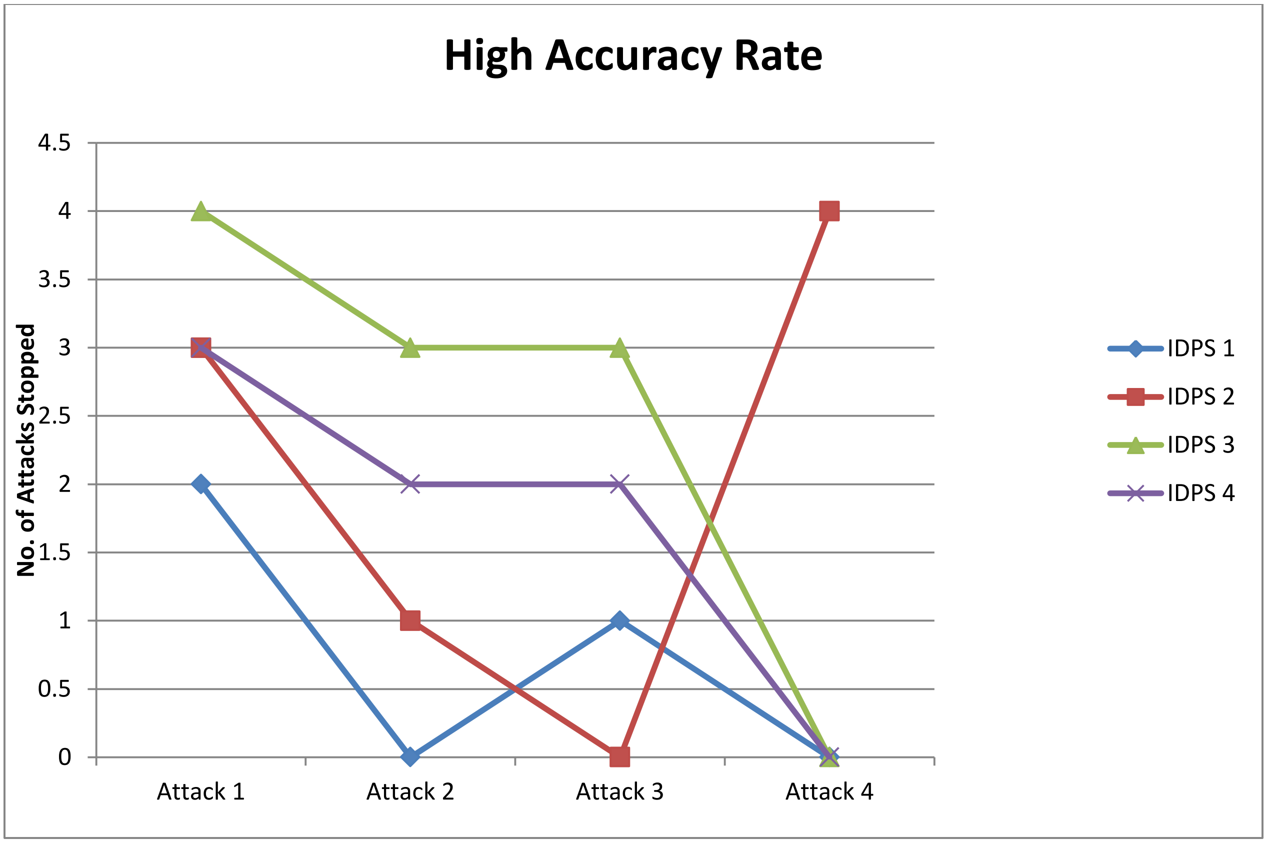

Figure 3.3. High accuracy rate.

5.3.4 Market share. Market share is the number of installations the IDPS has. A number of factors such as price, product/vendor name recognition, marketing, support, product effectiveness, and other affect how the IDPS is perceived in the market. We aggregated the information from public and private sources and came with Figure 3.4. Figure 3.4 shows how our test IDPS measure against one another in the IDPS market. IDPS 1 is the most installed IDPS in the world and it is the base of most commercial IDPSs [Snort]. IDPS 1's dominance of the market is due to the fact that it is free and has a very big community that provides free support. IDPS 2 has a very small market share due to how it works which reduces its broad appeal. IDPS 3 and IDPS 4 share about the same market share. Their market share is negatively affected by their cost and maintenance costs.

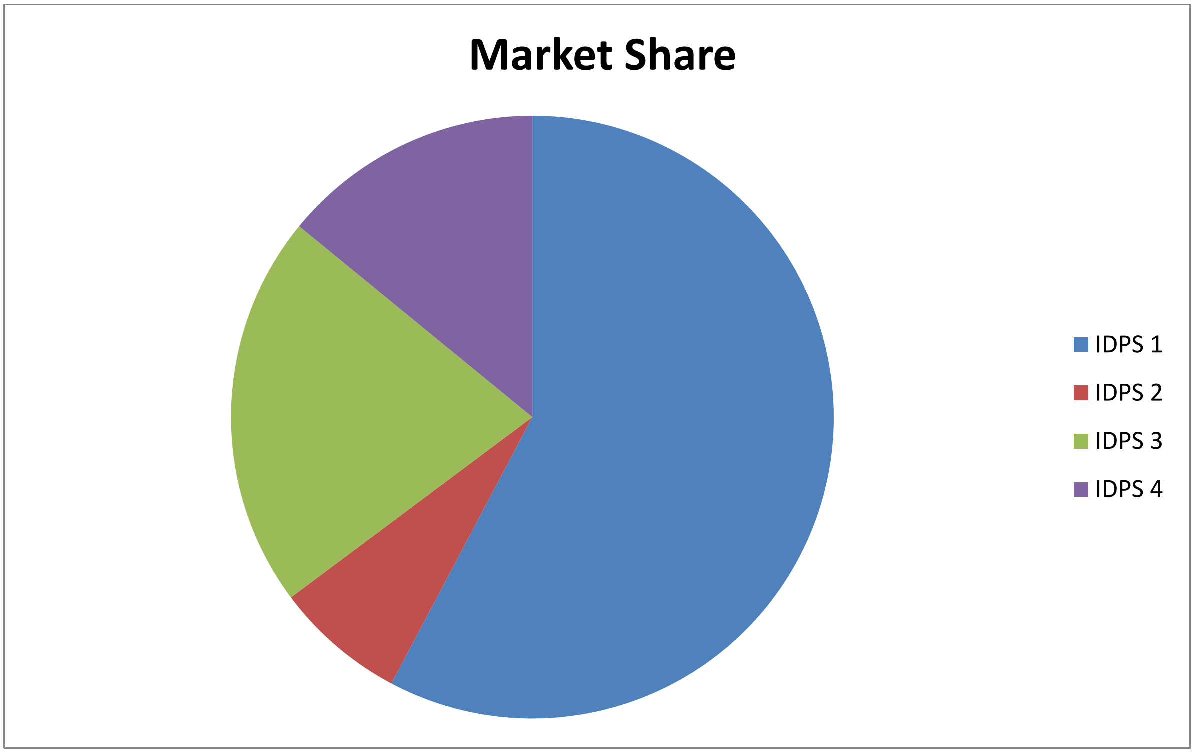

Figure 3.4. Market share.

5.3.5 Performance and overhead. An IDPS places some overhead on the monitored and protected network or computer systems. At minimum the IDPS has to have the same throughput as the network it monitors. If it has a lower throughput, it will become a bottleneck on the network. There is also a need to understand the difference between monitoring throughput and inspected throughput. It is our finding that there is a bigger gap between advertised IDPS monitoring speed and inspected speed. For example IDPS 3 has four ports that support 10/100/1000MBPS but only offers 50MBPS of inspected throughput. We used an evaluation copy of WhatsUPGold, a computer and network monitoring tool to take baseline performances of our test lab before, during, and after our evaluations. Using these baselines we came up with Figure 3.5 which shows how our test IDPS products performed and affected the test environment. We measured performance as a percentage of how long it took the IPDS to detect and prevent an attack. We also looked at how the IDPS get its updates and how complete are the updates. We found that just looking at the number of available signatures alone can be misleading. What is more important is the protecting the signatures offer against all known attacks and their variants and their relevance. For example if our environment has only one type of computer systems, we would prefer an IDPS that will automatically be aware of this or allow us to upload only the signatures for attacks that affect our systems rather than upload every signature they have. Uploading every signature leads to a bigger and less responsive detection times as the system has to processes the traffic against all the signatures. The overhead percentage was calculated by comparing the baselines of the monitored system. We compared how the system performed before and after the IPDS and also compared the baselines gathered during the attacks. Under normal operating conditions the overhead produced by the four IDPS products in our lab is negligible. Figure xx shows the numbers at pick load. IDPS behaves differently at pick performance and these differences need to be clearly understood. We found that IDPS 1 drops packets as it approaches its maximum inspected throughput and then fails open once it reaches it maximum. While IDPS 4 does not drop packets, instead it stops inspecting for predetermined period and then it starts inspecting again. IDPS 3 does not drop packets but just fails open and does not try to inspect until there is enough resources. Comparing IDPS3 and IDPS 4, it is clearly that IDPS 4 offers better performance. IDPS 1 offered the least performance and IDPS 2 is close to IDPS 3.

Looking at the overhead placed on the monitored system by the IDPS at its peak performance produces a better measure of how the system will behave. We measured overhead using WhatsUPGold during the complex attacks. During complex attacks we reduced the network throughput from 1 GBPS to 100 MBPS and placed background traffic on the network. This was done to fill up the network and push the IDPS to their maximum inspected throughput.

IDPS 2 place the least amount of overhead on the system. IDPS 2 performance here needs to be explained. Since IDPS 2 does not take any action when the attacks reach the machine, there was change in the performance monitor for the host that hosted IDPS 2 as the network traffic and attacks increased but and IDPS 2's resource use did not significantly spike as did IDPS1. IDPS 1 started dropping packets at 70 percent of its limit and placed the most overhead on the system. As IDPS 1 was struggling to keep up the load it also placed a significant load on the host that hosted it. IDPS 3 and IDPS 4 did not drop packets when the load increased instead they just let the packets go through. They both placed a small but noticeable overhead on the system. The difference between them was how they handled failures. IDPS 3 just failed open and did not inspect traffic until the attacks were over and the load was reduced on the network. IDPS 4 on the other hand continued to try and inspect traffic every minute until it could manage the load when the attacks started recede. This gave IDPS 4 a performance advantage over the other three IDPS products. How the IDPS products performed and the overhead they placed on the

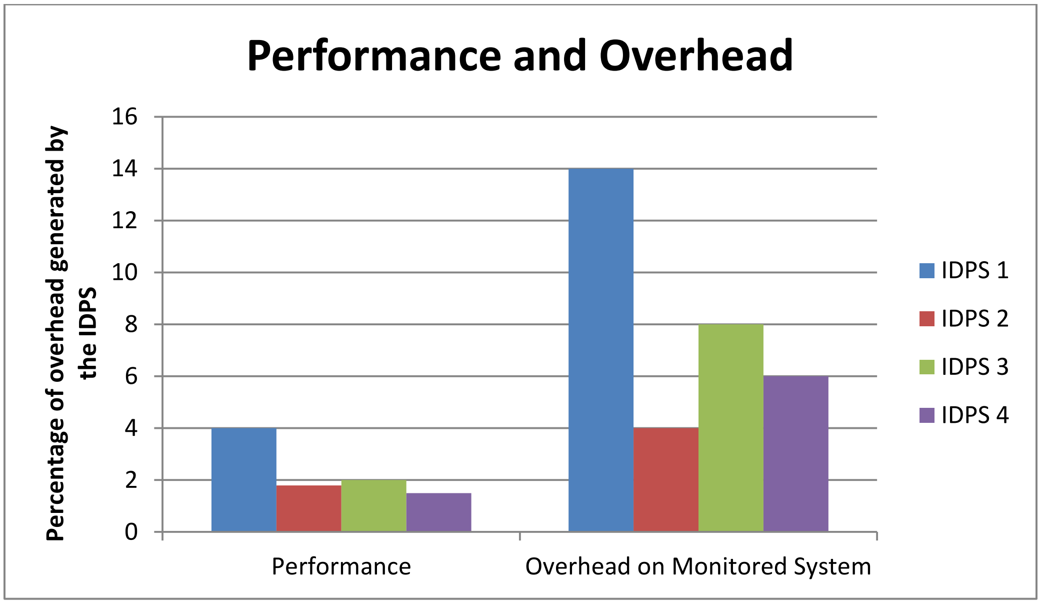

Figure 3.5. Performance and overhead.

monitored system is shown in Figure 3.5.

5.3.6 Maturity level. Maturity level is mainly measured by how long the product has been in the market and it is related to market share. We decided to expand our metrics to include more than just the years and acceptance. How much the IDPS has kept up with change and weather it continues to grow and accessibility were added to our formula. IDPS 1 has a sizable lead in this evaluation as portrayed by Figure 3.6. IDPS 1 is one of the earliest and the longest running IDPS product in the market and it continues to be used as the base for most commercial IDPS products [37]. IDPS 1 is an open source product that is freely available through a download to anyone through. IDPS 4 came in second followed by IDPS 3 and IDPS 2 is at the bottom of the list. IDPS 3 and IDPS 4 have progressed much better than IDPS 1 and IDPS 2 and have developed nice features, but these features come at an additional cost as an upgrade.

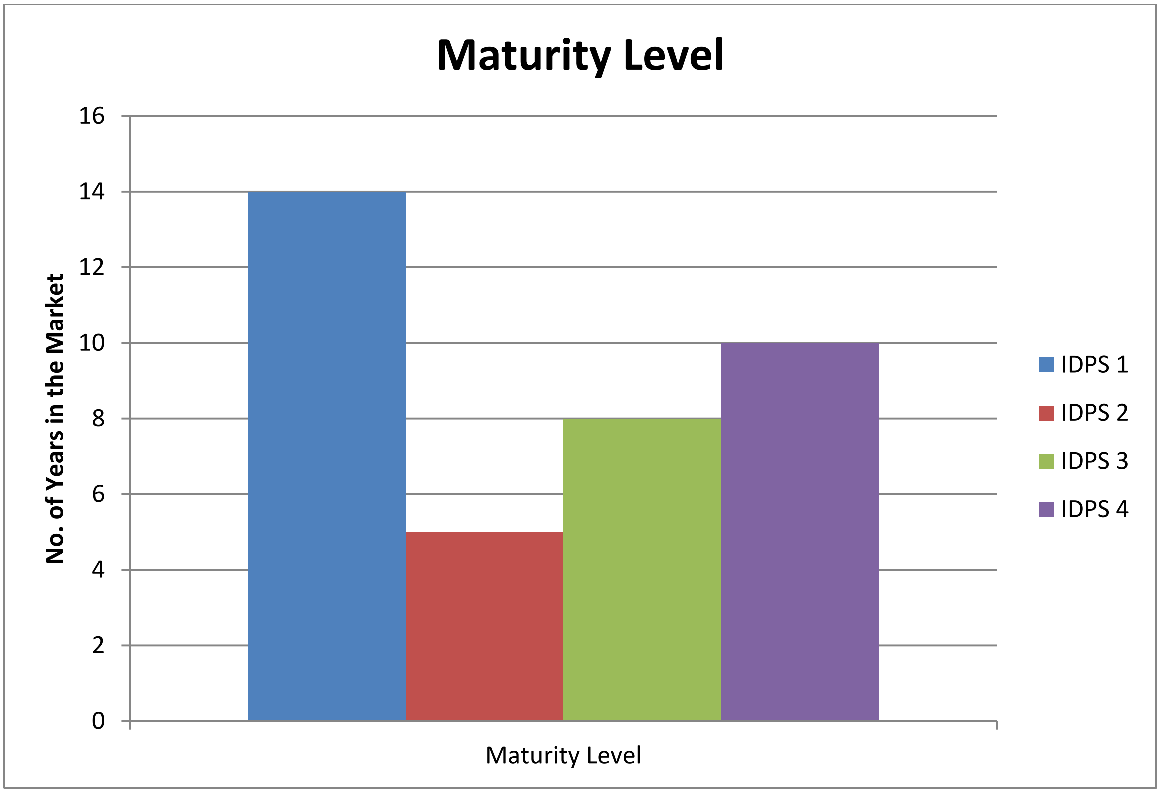

Figure 3.6. Maturity level.

5.3.7 Easy to use and configure. Ease of use and configuration are subjective and dependent on the environment and the skill set of the administrator. For our evaluation we polled some users of these IDPS and the compared that information with our experiences and the results are shown in Figure 3.7. IDPS 3 and IDPS 4 are appliances that come configured for a specific purpose and do not require a lot of skill to get them up and running. IDPS 1 and IDPS 2 are software based and require a host operating system to host them. IDPS 2 was much easier to install and configure when compared to IDPS 1. The installation on IDPS 1 was the most involved and required a number of extra steps before and after the installation on the host machine in order to get IDPS 1 working properly. IDPS 1 also required more time and skill to keep it updated with the latest signatures.

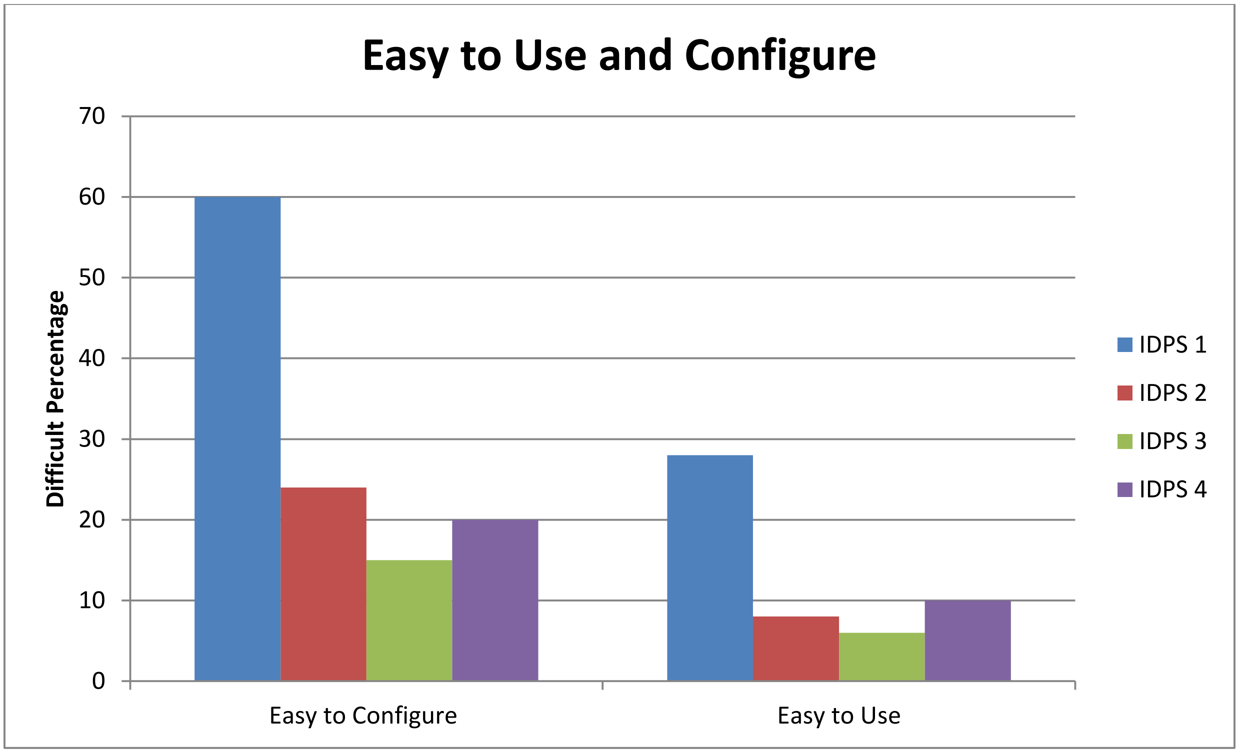

Figure 3.7. Easy to use and configure

IDPS 1 also took the most amount of time configuring, tweaking, and applying the signatures. IDPS 2 was a little involved too but not to the level of IDPS 1. IDPS 3 and IDPS 4 were the easiest install since all they required was power and there configuration was fairly simple. These appliance based IDPS products have a clear lead over the software one when it comes to installation and use as shown in Figure 3.7. IDPS 3 has the easiest web user interface that is just point and click and it also has a command line interface. IDPS 3 comes with most of its options already configured and can work straight out of the box. IDPS 4 takes just a little time to go through its options and make sure it is optimal before use.

5.3.8 False positives. False positives are alerts about attacks that are not correct. False positives pose no risks as they are results of the IDPS thinking that a non-attack is an attack. IDPS 2 produced that highest number of false positives alerts. This is in part due to the way it works. IDPS 1 was in second followed by IPDS 3 and IDPS 4 was at the bottom with the least number of false positive alerts. Figure 3.8 shows the false positives that we received during the evaluations.

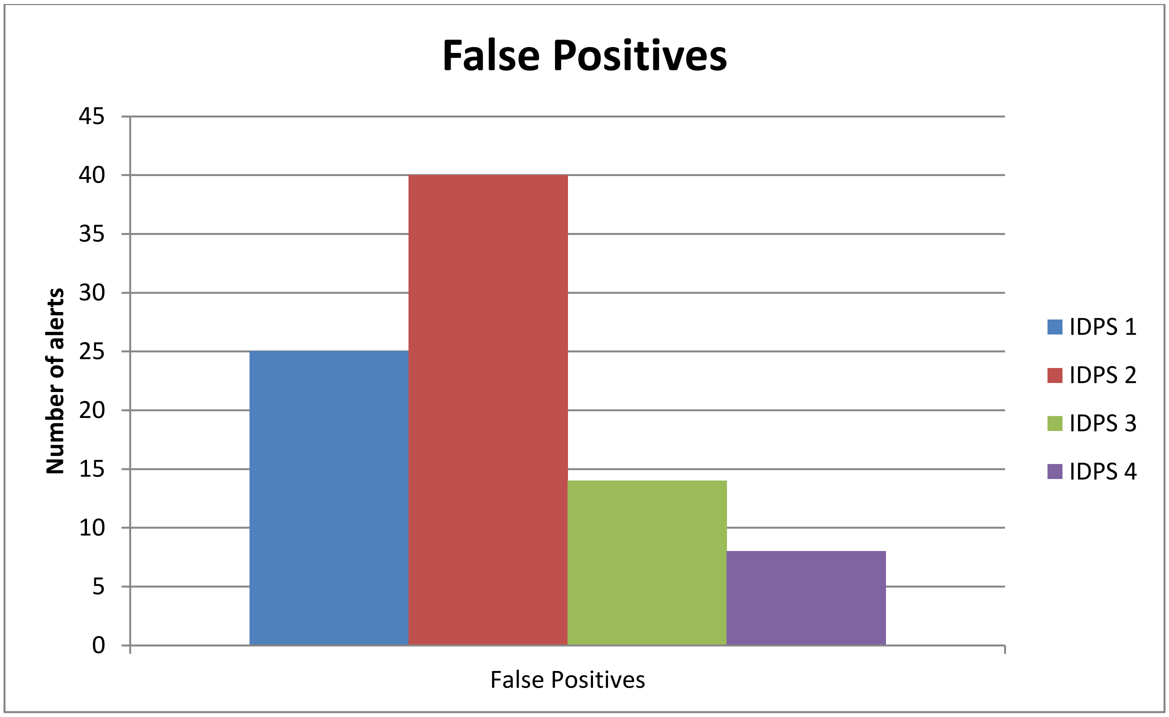

Figure 3.8. False positives.

5.3.9 False negatives. False positives are attacks that are successful and are not detected by the IDPS and a result no alerts are generated for them. False negatives are dangerous as they give a false sense of security. IDPS 2 had a high number of false negatives, but these numbers need to be further examined. Looking at the machines protected by IDPS 1 we discovered that all of our attacks made it to the machines but then IDPS1 stopped them or alerted when they tried to manipulate the targeted machines. IDPS 2 followed on the list, followed by IDPS 3 and IDPS 4 was at the bottom of this evaluation with the least number of false negatives. The results of the false positive evaluations are shown in Figure 3.9.

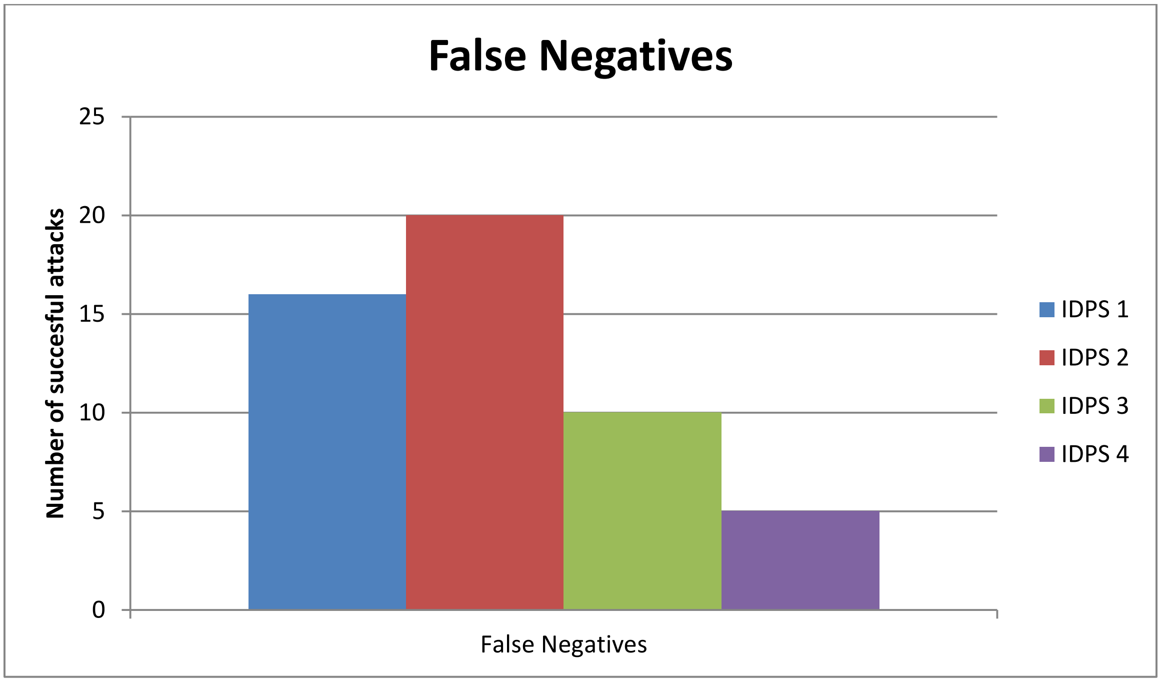

Figure 3.9. False negatives.

5.3.10 Cost and maintenance. We combined cost and maintenance as they are closely related. IDPS 1 topped the list on this evaluation since it is free and has no yearly maintenance fees that are associated with it. The only cost associated with it is the cost of server to host it. It runs and performs better on a Unix based operating system which further reduces its cost. There are a number of open source variants of Linux and in our lab we used SUSE Linux. We chose SUSE due to its ease of use and its large number of supported hardware. You could run IDPS 1 on a Windows based operating system but that will increase the cost and reduce the stability of the IDPS. Using Windows will also increase the chance of the IDPS been bypassed since there are so many ways to exploit a Windows operating system. IDPS 2 is not free but it is relatively cheap and has the same advantages as IDPS 1. IDPS 2 has a small yearly maintenance fee. IDPS 3 and IDPS 4 are very expensive to both purchase and maintain and they justify the cost by offering support, automated signature updates, leading research labs on threats. They also promise better detection engines and signatures that are continual been updated. In our evaluations these two produced better results and where easier to configure. Figure 3.10 portrays how these IDPS products compare to one another on cost.

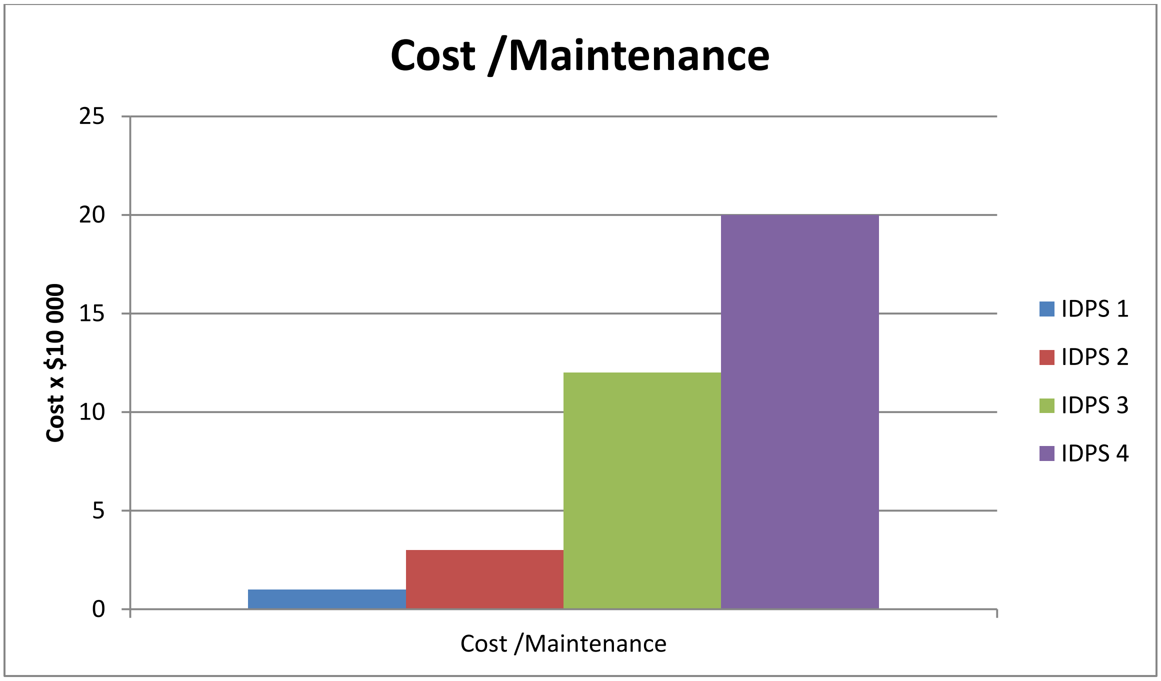

Figure 3.10. Cost/maintenance.

One of the cost disadvantages of IDPS 3 and IDPS 4 is that they are not scalable like IDPS 1 and IDPS 2. The appliance comes fixed as shipped and cannot be upgraded and the only way to upgrade these appliances is to buy the newer models that are much more expensive. The software based IDPS can easily be scaled by adding more resources such as memory, CPU, hard drive space, and network cards to the host server. Which is much cheaper and easy to do the getting a new appliance.

5.3.11 Protection against new attacks. When new attacks surface IDPS vendors quickly develops new signatures that detect and block these attacks, but this can take days, weeks, and even months. During this time systems protected by these systems will be vulnerable. Some IDPS such as IDPS 3 can detect and prevent new attacks without new signatures since they do not depend on signatures for detection. To evaluate how the test IDPS handled new attacks, we launched a variant of a Windows exploit that we downloaded from the internet. In this evaluation only IDPS 2 stopped a new attack and the other IDPS did not. Table 3.1 shows how our test IDPSs fared against new attacks. It is worth mentioning that after applying new signatures to IDPS 1 and IDPS4, they both detected and stopped the exploit. We could not clearly establish the time that lapsed between the exploit's discovery and the release of the signatures since the exploit has been around for a while and that type of work is outside of our focus. IDPS 2 employ anomaly based detection and that is why it performed so much better in this evaluation.

Table 3.1

Protection against new attacks and scalability

IPDS Products	**IDPS 1**	**IDPS 2**	**IDPS 3**	**IDPS 4**
Protection Against New Attacks	No	Yes	No	No
Scales easily	Yes	Yes	No	No

5.3.12 Scalability software based. IDPS products outperformed the appliance based one on scalability. IDPS 2 is the most scalable one since it runs on the monitored host and all it takes is to install on the new host. Even if it is centralized, to scale it you just add extra resources to the monitoring server. IDPS 2 is also easy to expand and grow since it requires no license and runs on servers which are fairly cheap and easy to expand as the environment grows. Our appliance based products are not as easy to expand and were equal bad on this evaluation. IDPS 4 is a little expandable since it has more ports than IDPS 1 but it ran out of processing power before it ran out of ports. For the appliance based IDPS the only solution to scale them is to replace them with new one which are much more expensive. Some vendors will offer trade-ins but that is not available on our test models. Table 3.1 shows how our test IDPSs scales.

CHAPTER 6

Conclusion and Future Work

This thesis addressed the misunderstandings of IDPS methodologies by explaining the four popular IDPS methodologies which are anomaly based, signature based, stateful protocol analysis based, and hybrid based detection methodologies. A method for evaluating and comparing these methodologies was developed and presented thereby simplifying the evaluation of IDPS methodologies and products that use these methodologies. This thesis also presented and discussed different ways to setup a test environment for evaluating IDPS products using open source tools Tomahawk and Wireshark. During the discussions three setups, the basic, medium, and advanced setups were developed. A way to create relevant PCAP files using Wireshark was also shown. Creating own PCAP files solves the lack of current data set problem and reduces the risk created by using questionable data sets from the internet. Using own PCAP files improves the evaluation results since current and representative network traffic will be used. The final phase of this thesis presented the analysis of the experiments that we conducted using the evaluation parameters, the setups, and test IDPS products. The experiments were a way to evaluate both the IDPS products and our evaluation parameters. The experiments also produced new data sets for use in IDPS evaluations.

This thesis was not focused on developing a new IDPS methodology or improving the ones that exit, instead it was focused on three things. The first thing was to identify and define four popular IDPS methodologies, second the objective was to offer a simple way to evaluate these methodologies, and third goal was to offer a simple way to evaluate and understand IDPS products. It is currently very difficult to evaluate and select an IDPS product and we offered a solution to this problem through our experiments. The parameters used to evaluate our test IDPS products can be used as a guide to help evaluate and understand what an IDPS offers and also validate its claims. For example most popular IDPS products claim to have a very high accuracy without explaining what they mean by high accuracy. Using our parameters an IDPS's accuracy rate can be broken down based on performance, overhead, scalability, usability and cost. We offered a way to evaluate vendor claims against user needs and we explained that lab results vary and based on the environment the experiments were conducted, for example almost all IDPS products will offer protection against known attacks and as a result the focus when considering an IPDS product is to look at the methodologies it uses and how it offers protection against new attacks. The other parameters that have to be understood before selecting an IDPS are cost, usability and scalability, most vendors tend to separate these parameters and present them in way that does not present the whole picture. This thesis explained how these parameters are interdependent, for example it does not matter how high the product is ranked if it is too expensive, difficult to use and does not scale well. Our evaluation parameters offers a simple and yet effective way to evaluate IDPS methodologies and products. Our setups solved the problem presented by lack of publicly available data sets for use in evaluating IDPS products by presenting ways one can create own data sets using publicly available tools. Using own data sets produces better results and reduces security risks posed by using data sets found on the internet.

Future work includes fine tuning and explaining our evaluation parameters and making them and our setups available to the public. We are looking at setting up a presence on the web for running more evaluations of both commercial and open source IDPS products and making our results public. The focus will be on evasion techniques, accuracy rates, and validating vendor claims and helping users better understand IDPS products and trends.

References

[1] Patcha, A., & Park, J. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends, Computer Networks. *The International Journal of Computer and Telecommunications Networking, 51(12),* 3448-3470.

[2] Bace, R. (1999). An introduction to intrusion detection and assessment for system and network security management. ICSA Intrusion Detection Systems Consortium Technical Report, 1999.

[3] Anderson, J.P. (1980). *Technical Report James P Anderson Co Fort Washington Pa*. Technical report, James P. Anderson Company, Fort Washington, Pennsylvania. Retrieved from http://www.citeulike.org/user/animeshp/article/592588.

[4] Sobh, S. T. (2006). Wired and wireless intrusion detection system: Classifications, good characteristics and state-of-the-art. *Computer Standards & Interfaces,* 28, 670– 694.

[5] Valeur, F., Vigna, G., Kruegel, C., & Kemmerer, A. R. (2004). A comprehensive approach to intrusion detection alert correlation. *IEEE Transactions on Dependable and Secure Computing,* 1, (3), 146-169.

[6] Wu, X. S., & Banzhaf, W. (2010). The use of computational intelligence in intrusion detection systems: A review. *Applied Soft Computing Journal,* 10, 1-35.

[7] Hoang, X. D., Hu, J., & Bertok, P. (2009). A program-based anomaly intrusion detection scheme using multiple detection engines and fuzzy inference. *Journal of Network and Computer Applications*, 32(6), 1219-1228.

[8] Elshoush, H. T., & Osman, I. M. (2011) Alert correlation in collaborative intelligent intrusion detection systems—A survey. *Applied Soft Computing* 11, 4349-4365.

[9] Shanbhag, S.; Wolf, T. (2009). Accurate anomaly detection through parallelism. *IEEE Network*, 23 (1), 22-28.

[10] Cannady, J., & Harrell, J. (1996). A Comparative Analysis of Current Intrusion Detection Technologies. *Pattern Recognition*, 96, 212-218.

[11] Bejtlich, R. (2004). The Tao of network security monitoring: beyond intrusion detection. Addison Wesley (p. 832). Addison-Wesley Professional.

[12] Brugger, T. (2007). KDD Cup '99 dataset (Network Intrusion) considered harmful. *KDnuggets News*, 18(4), 1-2.

[13] Scarfone, K., & Mell, P. (2007). Guide to Intrusion Detection and Prevention Systems (IDPS). *Nist Special Publication.* NIST.

[14] Pedro Garcı´a-Teodoroa, P., Dı´az-Verdejoa, E.J., Macia-Ferna´ndeza, G., & Va´zquezb, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenge. *Computers Security,* 28 (1-2), 2009, 18-28.

[15] Tsai, C.-F., Hsu, Y.-F., Lin, C.-Y., & Lin, W.-Y. (2009). Intrusion detection by machine learning: A review. *Expert Systems with Applications*, 36(10), 11994-12000.

[16] Denning, D. E. (1987). An Intrusion-Detection Model. *IEEE Transactions on Software Engineering*, SE-13(2), 222–232.

[17] Valdes, A., & Skinner, K. (2001). Probabilistic Alert Correlation. Recent Advances in Intrusion Detection, 54-68.

[18] Mukhopadhyay, I., Chakraborty, M., & Chakrabarti, S. (2011). A Comparative Study of Related Technologies of Intrusion Detection & Prevention Systems. *Journal of Information Security*, 2, 28-38.

[19] Patel, A., Qassim, Q., & Wills, C. (2010). A survey of intrusion detection and prevention systems. *Information Management Computer Security*, 18(4), 277-290.

[20] Weng, N., Vespa, L., & Soewito, B. (2011). Deep packet pre-filtering and finite state encoding for adaptive intrusion detection system. *Computer Networks*, 55(8), 1648-1661.

[21] Aydın, M. A., Zaim, H. A., & Ceylan, K. G. (2009). A hybrid intrusion detection system design for computer network security. *Computers and Electrical Engineering*, 35, 517–526.

[22] Ingham, K. L., & Somayaji, A. (2007). A methodology for designing accurate anomaly detection systems. *Proceedings of the 4th international IFIPACM Latin American conference on Networking LANC 07*, 139. ACM Press.

[23] Kruegel, C., Valeur, F., & Vigna, G. (2005). Intrusion Detection and Correlation: Challenges and Solutions. Evaluation, *Advances in Information Security,* 14, 122-136.

[24] Verwoerd, T. (2002). Intrusion detection techniques and approaches. *Computer Communications*, 25(15), 1356-1365.

[25] Noel, S., Wijesekera, D., & Youman, C. (2002). Modern Intrusion Detection, Data Mining, and Degrees of Attack Guilt. *Information Systems Journal*, 1-29.

[26] Javitz, H. S., & Valdes, A. (1991). The SRI IDES statistical anomaly detector. *Proceedings 1991 IEEE Computer Society Symposium on Research in Security and Privacy*. IEEE Comput. Soc. Press.

[27] Roesch, M., & Telecommunications, S. (1999). Snort – Lightweight Intrusion Detection for Networks. *Proceedings of the 13th USENIX conference on System administration*, 229–238. Seattle, Washington.

[28] Bashah, N., Shanmugam, I. B., & Ahmed, A. M. (2005). Hybrid Intelligent Intrusion Detection System. *Neural Networks*, 6, 23-26. Retrieved from http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1297598

[29] Depren, O., Topallar, M., Anarim, E., & Ciliz, M. K. (2005). An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. *Expert Systems with Applications*, 29(4), 713-722. Retrieved from http://linkinghub.elsevier.com/retrieve/pii/S0957417405000989

[30] Wang, S.-S., Yan, K.-Q., Wang, S.-C., & Liu, C.-W. (2011). A Hybrid Intrusion Detection System of Cluster-based Wireless Sensor Networks. Expert Systems with Applications, 38(12), 15234-15243. Retrieved from http://linkinghub.elsevier.com/retrieve/pii/S0957417411008608

[31] Dreger, H., Feldmann, A., Mai, M., Paxson, V., & Sommer, R. (2006). Dynamic Application-Layer Protocol Analysis for Network Intrusion Detection. *15th USENIX Security Symposium*, Retrieved from http://www.usenix.org/events/sec06/tech/dreger.html

[32] Mai, M., & Sommer, D. (2005). Dynamic Protocol Analysis for Network Intrusion Detection Systems. *informatikmaide*, 9. Retrieved from http://informatik-mai.de/files/Mai_DA.pdf

[33] Sourour, M., Adel, B., & Tarek, A. (2007). A Stateful Real Time Intrusion Detection System for high-speed network. *21st International Conference on Advanced Networking and Applications AINA 07*, 404-411. Retrieved from http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4220921

[34] Haines, J., Lippmann, R., Fried, D., & Zissman, M. (2001). 1999 DARPA intrusion detection evaluation: Design and procedures. Lexington MA MIT, (February). Retrieved from http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA38774

[35] McHugh, J. (2000). Testing Intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory. ACM Transactions on Information and System Security, 3(4), 262-294. doi:10.1145/382912.382923

[36] Mahoney, M. V. & Chan, P. K. (2003). An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection. *In Proceedings of the Sixth International Symposium on Recent Advances in Intrusion Detection*, 6, 220-237.

[37] Brugger, S. T. (2007). An Assessment of the DARPA IDS Evaluation Dataset Using Snort. *Electrical Engineering*, *1*, 1-19. doi : 10.1.1.94.674&rep=rep1&type=pdf

[38] Cardenas, A. A., Baras, J. S., & Seamon, K. (2006). A Framework for the Evaluation of Intrusion Detection Systems. *2006 IEEE Symposium on Security and Privacy SP06*, 0, 63-77). Retrieved from http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1624001

[39] Sommers, J., & Yegneswaran, V. (2005). Toward Comprehensive Traffic Generation for Online IDS Evaluation. *Development*, 12. Retrieved from http://pages.cs.wisc.edu/~pb/trident_final.pdf

[40] Corsini, J. (2009). *Analysis and Evaluation of Network Intrusion Detection Methods to Uncover Data Theft*, Master's thesis. Edinburgh Napier University, Edinburgh, UK.

[41] Athanasiades, N., Abler, R., Levine, J., Owen, H. & Riley, G. (2003). Intrusion detection testing and benchmarking methodologies. *First IEEE International Workshop on Information Assurance 2003 IWIAS 2003*, 63-72. doi : 10.1109/IWIAS.2003.1192459

[42] Sannella, M. J. 1994. *Constraint Satisfaction and Debugging for Interactive User Interfaces*. Doctoral Thesis. University of Washington, Seattle, WA.

[43] Wireshark. (2012), Wireshark, Retrieved from http://www.wireshark.org/docs/wsug.html

[44] Tomahawk. (2012), Tomahawk, Retrieved from http://tomahawk.sourceforge.net/

[45] Kayacik, H., Zincir-Heywood, A. N., & Heywood, M. I. (2005). Selecting Features for Intrusion Detection : A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets. *Proceedings of the Third Annual Conference on Privacy Security and Trust PST2005*, 3-8. doi : 10.1.166.7574

[46] Mudzingwa, D & Agrawal, R. (2012). A Study of Methodologies used in Intrusion Detection and Prevention Systems. *Proceedings of the IEEE SouthernCon 2012*. Retrieved from http:// ieeexplore.ieee.org/iel5/6190856/6196883/06197080.pdf

[47] Backtrack. (2012), Backtrack, Retrieved from http://www.backtrack-linux.org/

[48] LOIC. (2012), LOIC, Retrieved from http://sourceforge.net/projects/loic/

[49] Shahriar, M., Vahid, A., & Mojtaba, K. (2012). Effect of Network Traffic on IPS Performance. *Journal of Information Security*, 162-168.

[50] Antonelli, J.C. (2012). *Hands-On Network Security: Practical Tools & Methods, Security Training Course* [PowerPoint slides]. Retrieved from www-personal.umich.edu/~cja/HNS12/lectures/netsec-00-slides.pdf

[51] Husain, S., & Gupta S.C. (2011). Friend Features Extraction to Design Detection Engine for Intrusion Detection System in Mobile Adhoc Network. *International Journal of Computer Science and Information Technologies*, 2 (4), 1569-1573.

[52] Husain, S., Gupta, S. C., & Mukesh, C. (2011). Denial of Service attack in AODV & friend features extraction to design detection engine for intrusion detection system in Mobile Adhoc Network. *Computer and Communication Technology (ICCCT) 2nd International Conference*, 9, 292-297.

[53] Zanero, S. (2007). Flaws and frauds in the evaluation of IDS/IPS technologies. *First'07 Proc 19th Annual FIRST conference*. Retrieved from http://www.first.org/conference/2007/papers/zanero-stefano-paper.pdf