

Comparison between IDS and IPS

by Robert Weaver

FILE	14239_ROBERT_WEAVER_COMPARISON_BETWEEN_IDS_AND_IPS_78196_697228042.DOCX (1.75M)		
TIME SUBMITTED	21-APR-2016 07:47AM	WORD COUNT	12449
SUBMISSION ID	56172698	CHARACTER COUNT	73586



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

**Real Time Network Intrusion:
A Comparison between Intrusion Detection and
Prevention Systems**

By
Robert Weaver

Supervisor: Dr Carlene Campbell
Second Marker: Dr Kapilan Radhakrishnan

Project submitted as part of the requirements for the
award of BSc (Hons) Computer Networks

April 2016

DECLARATION

I, ... declare that I am the sole author of this Project; that all references cited have been consulted; that I have conducted all work of which this is a record, and that the finished work lies within the prescribed word limits.

This work has not previously been accepted as part of any other degree submission.

Signed :

Date :

FORM OF CONSENT

I hereby consent that my Project, submitted in candidature for the B.Sc.(Hons) Computing & Information Systems degree, if successful, may be made available for inter-library loan or photocopying (subject to the law of copyright), and that the title and abstract may be made available to outside organisations.

Signed :

Date :

Abstract

As the electronic age has evolved and the increased use of the Internet there has been a need to protect networks and data from both external and internal attacks. Today Organisations spend millions on security and training to mitigate risks but there is always a threat of network intrusion from both internal and external forces. This project will be looking in particular at Network Intrusion Detection/Prevention system SNORT and the challenges of creating SNORT rules to protecting against NMAP, DDoS and Hydra Attacks.

From research carried out a topology, a proposed topology (diagram of network devices) will be created on Windows 10 and Windows Server 2012 to test the effectiveness of protecting key and mission critical services. To protect these services rules will be created to protect against specific attacks to alert or drop malicious traffic depending on ruleset. To make the experiment more realistic a Pfsense an open source firewall will be setup to act as the network edge and block any outside traffic from entering the test bed to give inaccurate results.

Creating rules referencing from the Snort Manual and using PCAP pre-configured traffic compatible with Wireshark, Snort was able to analyse and produce a Log file showing the attacks making it easy and reliable method of testing Intrusion Detection. In conclusion, there is no 'Silver Bullet' a single solution to solving all security problems. Instead investing in a considered approach of planning and then implementing a deep level of security through firewalls and Intrusion Detection/Prevention devices and starting from the bottom up will provide a more complete solution in protecting data and services.

Table of Contents

List of Figures	6
Introduction	9
The Overall Aim	10
Objectives.....	10
Structure	11
Conclusion.....	12
Literature Review	13
Understanding Cybercriminals and Their Motivations	13
Security Threats of 2015	13
Zero Day Vulnerability.....	14
2015 Zero Day Attack Figures and Stats	15
Research into Zero Day Attacks	16
Stolen or Lost Data.....	19
Research in Protecting Data.....	20
Major Security Issues with Websites	21
Research into SQL attacks.....	22
Research into DDoS attacks	22
Example of DDoS and SQL injection.....	23
Intrusion Detection/Prevention Technologies.....	24
Research into using SNORT	24
False Alarms	25
Conclusion.....	26
Methodology.....	27
Design and Implementation.....	29
Introduction	29
Design.....	29
Requirement Specification.....	30
Topology.....	31
Why physical setup was not feasible	32
Virtual Setup with Cisco IP/Ds Device.....	34
SNORT	35
Firewall.....	36
Testing.....	37
Final Test Bed	37
Implementation	38

Explanation of Test bed	38
Example of false alarms	39
Hydra Attack	41
Testing Snort rules against Hydra Attack	42
Snort Output	43
Below are the Snort stream stats:	44
Snort Log files:.....	45
DDoS.....	46
Testing Snort from a DDoS attack.....	47
Start DDoS.....	47
Snort Output	47
Nmap.....	48
Testing NMAP port scan attack.....	49
Snort Output	50
Logged files	51
Conclusion.....	51
Evaluation	52
Conclusion and Recommendation	53
Reflection	57
References	59

List of Figures

Figure 1: New vulnerabilities Zero day attacks

Figure 2: Zero Day Attacks

Figure: 3 Critical Websites

Figure 4: Graph representing the figures from 2013 to 2015 on the number of total breaches

Figure 5: Graph shows the number of mega breaches from 2013 to 2015

Figure 6: Graph showing the average identities exposed per breach

Figure 7: Graph showing the median identities exposed per breach from 2013 to 2015

Figure 8: The most frequently exploited and targeted companies in 2014 and 2015.

Figure 9: Research Onion

Figure 10: Network detection topology using Cisco devices

Figure 11: Network Prevention topology using Cisco devices

Figure 12: Cisco IME

Figure 13: Cisco IPS in GNS3

Figure 14: Cisco IPS configuration

Figure 15: Snort Topology

Figure 16: Final Test Bed

Figure 17: Final Test Bed

Figure 18: Snort

Figure: 19: False Alarms

Figure 20: False Alarms

Figure 21: Windows 2012

Figure 22: Pfsense

Figure 23: Pfsense

Figure 24: Wireshark FTP

Figure 25: Wireshark FTP

Figure 26: Hydra

Figure 27: Running Hydra with ruleset

Figure 28: Hydra Output

Figure 29: Alerts

Figure: 30 Alerts

Figure 31:

Figure 32: Notepad Log

Figure 33: DDoS

Figure 34: DDoS

Figure 35: DDoS

Figure 36: DDoS

Figure 37: Wireshark Nmap

Figure 38: Wireshark Nmap

Figure 39: Nmap

Figure 40: NMAP rules

Figure 41: Nmap Outcome

Figure 42: Nmap Outcome

Figure 43: Nmap Logged Files

Figure 44: Nmap Log

Introduction

As the electronic age has evolved and the increased use of the Internet there has been a need to protect networks and data from both external and internal attacks. Today Organisations spend millions on security and training to mitigate risks but there is always a threat of network intrusion from both internal and external forces. So what is 'network intrusion?' Network Intrusion is an illegal attempt to breach a network do damage to network devices, infrastructure or to comprise integrity of data.

To protect against network intrusion organisations, they have to integrate a deep level of security from network edge down to users. Standard practise is to use a mixture of security technologies which complement each other as there is no 'silver bullet' solution.

Technologies such as firewalls and Intrusion Detection/Prevention are situated near resources to provide deep packet inspection to detect and stop any attacks against key or mission critical services. These technologies are useless unless they are properly configured and tuned with the right ruleset and monitored constantly by security personnel. In this chapter the project will be introduced by examining the overall aim, objectives and a breakdown of the contents of each individual chapter on what to expect to be included.

This project will specifically be looking at protecting data and network devices by producing a comparison between Intrusion Detection/Prevention by using SNORT an open source platform. To detect network attacks SNORT will use a rules which can be modified to alert or drop attacks (IDS and IPS) modes, the log files will then be stored using Barnyard2 and viewed using BASE a GUI interface for SNORT. By comparing the two methods of Intrusion Detection/Prevention the overall aim is to produce recommendations for businesses on the placement, configurations, positives and negatives of these systems. Each will be tested in a

Virtual box Environment connected to virtual machines with Kali Linux conducting three network attacks and from these attacks a comparison will be made. SNORT will also be placed on a Windows Server 2012 to emulate protecting mission critical services and will be tested again to assess the effectiveness.

The Overall Aim

The overall aim of this project is to assess the effectiveness of Intrusion Detection/Prevention Technologies and how they can be tuned to be more effective in detecting and preventing attacks. The secondary aim is to deliver a detailed guide on different types of network intrusion and detection systems used today in organisations with important emphasis on following industry standard practises by looking at placement, configuration and positives and negatives of each technologies. The outcome of this project is to gain industry knowledge on network intrusion and on how to secure a network using appropriate devices and standards.

Objectives

In order to achieve the overall aim of the project a series of objectives must be met. Below are the objectives:

1. Create a Virtual box network using SNORT, Kali Linux and Windows Server 2012
2. Configure SNORT rules in both IDS/IPS modes to monitor for PING, NMAP and DDoS attacks.
3. Produce a comprehensive comparison between SNORT IDS/IPS rule set.
4. Analyse the results.

5. Make recommendations for suitability of IPS and IDS for small, medium and large networks.

Structure

In this section of the project the structure will be discussed. The project is split up into three sections:

- Theoretical
- Practical
- Evaluation

In theoretical section of the project refers to teaching and learning approaches both inside and outside the project. The second part practical will contain the planning, design and development of testing SNORT and the analysis, and the results of the experiment. The third and final part of this project is the evaluation of findings and comprehensive comparison of both a simulated method and a physical practical method.

Below is a breakdown of chapters which will be included in this project:

- **Introduction:** Illustrates an overview of the report. It details the aims and objectives that once achieved will bring success to this research.
- **Literature Review:** In this chapter research around network intrusion and IPS/IDS technologies will be critically evaluated and compared with each other a highlight the differences between current research literatures. Also Symantec annual report will be analysed in depth to give figures and stats on network intrusions and main threats of 2015.

- **Methodology:** This chapter will explain in detail the procedures in order to create a solution to the idea trying to be accomplished and show readers how to replicate this study.
- **Design and Implementation:** This chapter is split into two parts Design and Implementation. In Design section will talk about the design considerations and how the experiment will be carried out and how contributed to the overall experiment. The Implementation section will discuss how experiment was configured with screenshots as well the outcomes of each test.
- **Evaluation-** This chapter will talk about evaluate results collected from the experiments and if they achieved the overall aim of the project.
- **Conclusions and Recommendation-** This chapter will sum up project from each chapter and provide any future lines of research.
- **Reflections-** The final part of the project is to reflect on the experience of writing a lengthy academic piece of work and it has impacted on studies and person.

Conclusion

This chapter has discussed several things. At the beginning of this chapter it looks at aims the project wishes to achieve and also introduced the overall main aim of testing the effectiveness of SNORT an Intrusion Prevention/Detection system and how it can be applied to a live network. Following on from this both of the aims and objectives were listed in order to test SNORT. Then the breakdown of each chapter on what they will contain.

Literature Review

In second section of this project a series a literature around the chosen topic will be analyzed, in this case to keep research relevant topics will analyze and centered around this year 2016 and last year 2015. The section will be focusing on two parts, Security Threats of 2015 and Intrusion Detection/Prevention technologies.

Understanding Cybercriminals and Their Motivations

As the electronic age has evolved and the increased use of the Internet there has been a need to protect networks and data from both external and internal attacks. Today organisations spend millions on network infrastructure and training to mitigate risks but there is always a threat of network intrusion from both internal and external forces. But why do these attacks take place? Perhaps for financial gain, Industrial or Political espionage, Military Warfare or maybe for Activism or Hacktivism [1]. The reasons for cybercrime may differ but the outcome, the objectives all aim to disrupt, bring down or exploit data, devices or software.

Security Threats of 2015

Before carrying out security training or implementing appliances to a live network it is important to be educated on the threats out there which target networks and data integrity. To understand these threats data must be analyzed on the most frequent and damaging attacks to provide a picture on they operate how they exploit vulnerabilities within a network. The year 2015 was notorious for network intrusion and the comprise of data which cost companies heavily financially and in customer trust. The Internet Security Report 2016 will be the main focus in this section. The report produced by Symantec [2] every year documents these vulnerabilities from over 157 countries and territories through Symantec products and services as well as other third party data. The report starts off highlighting by stating the five key findings, below they are bulleted:

- A new Zero-Day vulnerability was discovered on average each week
- Over half a billion personal records were stolen or lost in 2015

- Major security vulnerabilities in three quarters of popular websites

Each of the above will be the basis for research in depth and supported with white papers, journal articles and books relating the subject.

Zero Day Vulnerability

The first key point discussed from the Symantec report was that a zero-day attack was being discovered on average each week. But what is a zero-day attack? According Leyla Bilge and Tudor Dumitras who wrote a white paper 'Before We Knew It, An Empirical Study of Zero-Day Attacks in the Real World' A zero-day attack is a cyber-attack which exploits a vulnerability that has not been disclosed publically [3]. Zero-day attack can be accomplished using many different forms but mainly: worms, viruses, Trojans [4] but the most effective attacks to avoid detection are polymorphic worms which have the ability to evade, defenses, scanning appliances and replicate according to R. Kaur and M. Singh [5].

Worryingly there is no way to defend from a zero-day attack because it remains unknown to the software developer and therefore it cannot be patched. This also comprises anti-virus or intrusion detection/prevention software and hardware as they use signatures to detect intrusions and by not being known signatures cannot be written to combat the threat. L. Bilge and T. Dumitras go on to say that 'cyber criminals target Fortune 500 companies with millions of consumer PCs around the world and for this reason a new vulnerability has a potential market value of \$5,000 to \$250,000' [3]. Linking back to start of this section different motives cyber-attacks were discussed, in this case discovering a vulnerability in a consumer based software has a monetary value which could be used to blackmail the parent company or customer thus decreasing public relations with that company.

2015 Zero Day Attack Figures and Stats

To show the extend of the problem, there are a series of graphs below highlighting the numbers of new vulnerabilities, zero-day vulnerabilities, websites with vulnerabilities and percentage of which were critical all taken from the Symantec report. Below are the graphs:

The first graph below shows general new vulnerabilities:



Figure 1: This graph shows a decrease from 2013 to 2015 but there is still over 5,585 total vulnerabilities being detected in 2015.

The second graph shows amount of zero-day vulnerabilities:



Figure 2: This graph shows a 125% rise from 2014 to 2015 in zero-day attacks

The third graph show both Websites which where scanned in resulting to be vulnerable and also the percentage of which were critical:



Figure 3: This graph shows that 78% of scanned websites where vulnerable and 15% of 78% believed to be critical.

As you can see there are still a high number of vulnerabilities within existing software and websites in 2015 which can lead to a zero-day attack. Symantec is a well-respected company with a large consumer base both home and commercial users and has produced reports like this for several years with high accuracy due to them being imbedded in the security industry. It is important to them publish accurate data as they operate in a very competitive industry.

Research into Zero Day Attacks

There are many examples of academic research into zero day attacks. In this section several examples will be discussed and critical examined. The piece of academic literature is written by Massimiliano Albansese, Sushil Jajodia, Anoop Singhal and Lingyu Wang titled 'An Efficient Approach to Assessing the Risk of Zero Day Vulnerabilities'. In this paper they discuss a more suitable approach to detecting Zero Day Attacks which are more scalable compared to existing methods [7]. In this paper they highlight that zero day attacks can be easily modeled but are difficult but producing security defenses against them are difficult to their unappreciable nature. The Paper then discusses current predicting methods to detect zero day attacks and mentions the 'k-zero-day safety metric' [7]. The downside of this metric system is that it is not scalable and not feasible for large networks or complicated software. The aim of this research piece was to effectively re engineer the 'k zero-day metric' to make it more scalable for large networks and software houses. They did this by changing how the metric attack graph works by introducing a set of polynomial algorithms which instead located attacks more efficiently due to the way in search for them. This piece of research is needed to combat potential attacks in the future and help software and networks become more secure and protect against the threat of zero day attacks.

Another form of method in detecting zero day attacks is a 'Contextual Anomaly Detection Approach'. This research was carried out by Ahmed AlEroud and George Karabatis from the Department of Information Systems at the University of Maryland [8]. This paper describes the use of contextual misuse approach combined with an anomaly detection technique to detect cyber-attacks. They describe 'contextual misuse detection' as a system to place zero day attacks in some sort of category to make easier to identify. Like with the research conducted in 'Efficient Approach to Assessing the Risk of Zero Day Vulnerabilities' they design an algorithm to make protecting networks and software easier. In this case they describe using the 'One Class Nearest Neighbour' algorithm. Aim of this algorithm is to

‘detect any attacks that are relevant and specific to particular context. This means they can limit the number of false positives detected by an Intrusion Detection System which makes detecting actual attacks have a higher success rate. This research is vital in keeping appliances like Intrusion Detection systems on the front line against cyber criminals and increasing the effectiveness of security devices up to date in preventing attacks.

The research paper also focusses on the effectiveness of detecting zero day attacks but in a completely different method. In case a more proactive approach is taken to detect and learn about zero day attacks. In the paper ‘Detecting and Analyzing Zero-day Attacks using Honeypots’ Constantin Musca, Emma Mirica and Razvan Deaconerscu analyses the effectiveness of using a honeypot system to learn about zero-day attacks [9]. They start by saying networks are ‘overwhelmed by self-propagating malware such as worms, viruses and Trojans’ and the number of zero day attacks are increasing by each week. The aim of this project is to present methods for isolating malicious traffic by using a honey pot system in order to generate signatures for Snort. Snort will be discussed later in this project and used as the practical element. In order for signatures to be generated, C.Musca, E. Mircia and R.Deaconescu setup a virtual network and used to Snort to gather as much data as possible to log in on a remote server through a safe connection. This research is important for both security industry and academics who are learning the capabilities of Snort as it keeps signatures and methods of deploying the Intrusion detection system up to date with the latest information on zero day attacks. Keeping signatures up to date is vital for continued research in the cyber security field as there is a constant battle between cyber security officials and hackers as they try to outsmart each other. This research is admirable because it takes a more proactive approach in bringing in attackers in by creating a honeypot system

The next research supports ‘Detecting and Analyzing Zero Day Attacks’ as it also takes a proactive approach in developing and improving Network Intrusion Systems. The paper ‘Intrusion Detection System using Honey Token based Encrypted Pointers to Mitigate Cyber Threats for Critical Infrastructure Networks’ written by Muhammed Kamran Aisf and Yahya Subhi Al-Harhi [10] takes a less conventional method in detecting and stopping zero day attacks. In the paper it states that ‘the scale of damage that could be done by well-planned cyber-attacks is enormous’ [10]. This paper moves away from the norm and looks for a more inside the book approach for detecting zero-day attacks. M.Aisf and Y. Al-Harhi use a ‘novel’ approach of Honey token based encryption pointers to protect network infrastructure with aim of ‘trapping’ the attacker [10]. The system does this by dividing itself into four different nodes based on level of vulnerability and are protected by a IDS appliance. The IDS

appliance has a range of encryption schemes which include AES-126 to 256 and were attacked. M.Asif and Y.Al-Harthi then attacking these nodes and analysed the results in form of the IDS detecting false and true positives. Based on this performance M.Asif and Y.Al-Harthi then tested in a more realistic manner by putting 64 nodes under attack to represent a Denial of Service attack. This is a more practical method of testing IDS technologies as it represents a real world scenario on a live network. Finally from this corresponding data M.Asif and Y.Al-Harthi were able to reverse engineer to find where the attack was coming from and find the software bug which was causing it. From the research looked at so far this paper was the first to show a method of re-engineering a zero-day attack to find the root cause. This is very beneficial to the cyber security community as it provides a method to hunt, track down and find both the cause and bug the attacker has been trying to take advantage of. Another benefit of being able to reverse engineer the attack is to see time stamps on where the attacker has gained entry, exploited and move onto other chinks in the armor of the defense in a network. This helps cyber security professionals who specialize in disaster recovery as it gives them a time line of the attack and what data and services could be potentially affected.

In this section of the literature review stats and figures on the year 2015 have been analysed by Symantec and represented in a graphic manner with the use of charts. A focus group of research has been critically analysed by explaining the methods and logics behind their thinking, results and how well they performed practical experimentation through normal and unconventional ways. It is important to increase the amount of research dedicated to improving signatures and rulesets of Intrusion Detection systems as the fight against cyber criminals will only get harder in the future with the exploitation of more advanced and focused attacks directed at today's networks and data.

Stolen or Lost Data

The second major threat presented in the Symantec Cyber Security Report is stolen or lost data. According to the report at the end of the year 2015 over 191 million records were exposed [2] with a record setting 9 mega breaches (a breach with over 10 million records exposed). Furthermore, the total number of exposed identities jumped to over 429 million an increase of 23 percent but more concernly companies not reporting the number of breaches has increased by 85% and according to Symantec this pushes lost or exposed data up into the 1 billion bracket [2]. It is shocking that companies are reporting exact figures as the cyber security field needs to be transparent with both attacks and research to continue to develop. Below are some graphs that highlight problems with security breaches in the year 2015:



Figure 4 Above is a graph representing the figures from 2013 to 2015 on the number of total breaches



Figure 5 Above is a graph show the number of mega breaches from 2013 to 2015



Figure 6 Above is a graph showing the average identities exposed per breach



Figure 7 Above graph show the median identities exposed per breach from 2013 to 2015. As you can see from the graphs there has been a constant rise in the number of breaches within the past 3 years. This is worrying because ransomware is also increasing and maybe a direct correlation between the two stats.

Research in Protecting Data

In this section of the literature review research surrounding protecting personal data will be reviewed and critically analyzed.

In the first piece of research surround the topic area, Fara Yahya, Robert J Walters and Gary B Wills discuss 'Protecting Data in Personal Cloud Storage with Security Classifications' [12]. In the paper they discuss the increase importance of protecting cloud data due to the ever use of cloud services. Protecting data from unauthorized access requires a deep level of security and need some sort of standardized guidelines/framework for everyone to follow so there is a level field of security. In 'Protecting Data in Personal Cloud Storage with Security Classifications' it discusses a need for guidelines and standard bodies to provide security data to provide less user convenience, decreasing availability and the need for robust infrastructure' [12]. The paper proposes different levels of security threat between Server to Low based on threats like brute force, inconsistent encryption, hardware failure, Malware, DDoS and Man in the middle attacks. To give a framework the paper suggests focusing on three areas, Authentication, Authorization and Encryption. Is along with security infrastructure such as firewalls and deep level packet inspection via Intrusion Detection/Prevention systems. A long with physical devices protecting data, the paper also suggests a three tier security protecting those devices from damage. Security technologies such key code, biometric scanner and smart are all suggested. This piece of research is not directly linked to how network intrusion detection work/operate but providing secure access for those devices in a server room prevents tampering or destruction of the security devices which will leave the network

vulnerable to attack. It is important to provide deep level of security to all factor all levels of a network right from end users to the network edge itself.

Major Security Issues with Websites

The third major security threat highlighted by Symantec in their 2015 security is Major Security issues with websites. Symantec state that trusting well known websites will prevent cyber-crime. But this is not true, more than 75 percent of websites have unpatched vulnerabilities while 15 percent of those deemed to be critical [2]. This results in them becoming prime targets for hackers as they are normal gateways to finical reward which was discussed in the ‘Understanding motivates of cyber criminals’ section of this chapter. Symantec gives a diagram on threat percentage likelihood based on organization type. Below is the diagram:

	2015 Top 10 Most Frequently Exploited Categories of Websites	2015 Percentage of Total Number of infected Websites	2014 Top 10	2014 %
1	Technology	23.2%	Technology	21.5%
2	Business	8.1%	Hosting	7.3%
3	Search	7.5%	Blogging	7.1%
4	Blogging	7.0%	Business	6.0%
5	Dynamic	6.4%	Anonymizer	5.0%
6	Educational	4.0%	Entertainment	2.6%
7	Domain Parking	3.2%	Shopping	2.5%
8	Entertainment	2.6%	Illegal	2.4%
9	Shopping	2.4%	Domain Parking	2.2%
10	Illegal	2.1%	Virtual Community	1.8%

Figure 8 Above are the most frequently exploited and targeted companies in 2014 and 2015.

The main threat named in the report is SQL injection. SQL injection attack consists of inserting or “injecting” a SQL query in an input data from an application [14]. A successful attack can insert, update and delete records and to emorus amounts of damage. Another side

to a SQL attack is that it can start a situation of a blackmail scenario potential costing an organization million. The section below will look at SQL injection research and how to secure against it. Another attack against websites and server is a DDoS attack

Research into SQL attacks

The first piece of literature in SQL attacks and preventing them is a research paper 'SQL injection is Still Alive: A Study on SQL injection Signature Evasion Techniques' by Amirmohammad Sadeghian, Mazadak Zamani and Suhaimi Ibrahim [15]. To protect from SQL attacks it is important to understand how they work but also they evade intrusion detection systems. The paper states that SQL injection is the highest rank web base attack and states the most common way to defend against it is implement a web based firewall. The firewall uses a signature based technique to scan each query but can be outsmarted if the attacker has a good knowledge of SQL [15]. The paper suggests to implement a network intrusion detection/system and web based firewall together to along communication will website developers to provide security. This paper highlights a key point in research into intrusion detection/prevention systems that there is no silver bullet solution to protecting against all types of attacks and all departments must be transparent with each other to provide constant security coverage either through updates in signatures or ruleset, website patches or provide secure access to storage data and devices. The research carried out in this project provides a clear reason why Intrusion detection/prevention are not enough on their own to provide complete security protection against every attack. Planning and being proactive in a security sense gives the best overall defense but if the worst was to happen it a breach in a network or data planning provides disaster recovery to limit the amount of damage is done.

Research into DDoS attacks

The second part of this section will be dedicated to Denial of Service research. Denial of service equally can effect websites and network services and as mentioned in 'Protecting Data in Personal Cloud Storage with Security Classifications' providing stable secure infrastructure stops user inconvenience and decreasing availability [12]. The first research paper looking into defending against denial of service attacks is a paper written by Andrew Carlin, Mohammad Hammoudeh and Omar Aldabbas titled 'Defense for Distributed Denial of

Service Attacks in Cloud Services' [18]. The paper critically reviews the most recent intrusion detection and prevention systems to mitigate potential DDoS attacks. A.Carlin, M. Hammoudeh and O.Aldabas first start off in discussing the vulnerabilities. 'Openness of the Cloud and large amounts of data makes them tempting targets for attacks'. Linking back to very first section the motivations of cyber attackers was discussed either through financial gain, Industrial or Political espionage, Military Warfare or maybe for Activism or Hacktivism [1]. The paper then moves to discussing the 'classification and challenges' of Intrusion detection devices. There four types of intrusion detection they are: host based HIDS which monitors and analyses log files, security access and user login data, Network based IDS (NIDS) monitors IP transport layer headers with behavior being compared with signatures, Hypervisor based IDS (HyIDS) monitors and analyse communication between VMs and Distributed IDS (DIDS) consists of all three [19]. After the paper lists the different types of Intrusion detection and prevention A.Carlin, M. Hammoudeh and O.Aldabas then discuss the pitfalls of intrusion detection/prevention systems. 'They generate false alarms' [19], no uniform standard or metric for evaluating an IDPS which made lead to misleading information. The paper then includes in saying that IP/Ds alone are not enough to defend and protect a network and due to their high generating in false alarms. This piece of research provides a critical review and standse highlight the pitfalls and downsides implementing ID/Ps in a cloud and network environment. It makes the reader think carefully of placement and testing before applying an intrusion detection/prevention and test in thoroughly before implementing it to a live network.

Example of DDoS and SQL injection

To be a victim of a SQL Injection and DDoS attack can be very costly due potential black mailing and damaging to a company's reputation. On Wednesday 21st October 2015 TalkTalk a British Telecoms company was attacked with both a DDoS and SQL Injection Attacks. Over 21,000 unique bank account numbers and sort codes were stolen, 28,000 obscured debit and credit card details. 15,000 dates of birth and 1.2 million names and addresses, email and phone numbers [7]. These attacks severely damaged the reputation of TalkTalk as it was the third time in two years (December 2014 India Based Phishing Calls in February 2015 and affected by the attack on Carphone warehouse) [8]. This lead to a large up cry and an investigation from the British Government.

Intrusion Detection/Prevention Technologies

This section is part 2 of the literature review which will focus solely on both Intrusion Detection/Prevention technologies and look at placement, configuration of rules and pros and cons of implementing intrusion detection/ prevention systems.

Research into using SNORT

In this section will look at research which used SNORT a free open source intrusion detection platform. The first piece of research is 'Performance evaluation comparison of Snort NIDS under linux and Windows Server' by K.Salah and A.Kahtani [25]. In the paper they explore and evaluate the performance of Snort on different platforms Windows Server 2003 and Linux. To measure the performance of Snort they test the servers under two different load conditions, normal and malicious and look at throughput and packet loss [23]. To find test SNORT they looked the use of RAM and CPU usage by tweaking for a 'fine grain control'. On the Windows platform both K.Salah and A.Kahtani tweaked the processor scheduling and on the Linux platform they adjusted budget configurable parameter used by Linux kernel packet reception mechanism [23]. The outcome of the experiment was that Windows processor scheduling provided little to no difference compared to Linux and that under heavy load conditions Linux platform outperforms Windows Server 2003. But under medium load Windows Server exhibited better performance under moderate to normal load conditions. This research is important to the deploying of Snort IDS to protect servers in a network. Both Linux and Windows show advantages under different loads and how tweaking RAM and CPU usage impacts the performance. This gives a basis for network security administrators to apply their Snort configurations on either a Linux or Windows platform.

The next research paper 'Improving Intrusion Detection System Based on Snort rules for Network Probe Attack Detection' by Nattawatt Khamphakdee, Nunnapus Benjamas and Saiyan Saiyod [24]. In this paper it aims to improve Snort Rules and reduce false alarms. The team tested Snort by using two different methods, the data set from MIT-DAPRA 1999 which contain normal and abnormal traffic and used Wireshark shark to analyze the traffic. The second test used attack data set in which also they applied Wireshark to further analyses the traffic. The outcome of the experiment was that they were able to improve Snort ruleset and have a 100% detect rate of the network probe attacks. Paper concludes in saying snort

rules can help administrators quickly analyse the patterns of attack [24] but highlight that the Snort rules must be updated regularly to keep up with attacks because attackers will increase the complexity of their attacks which could result in a large scale damage to data integrity and network devices. This paper is important for the development of the Snort ruleset to provide the security community and academics the necessary tools to learn and implement security technologies to their home or commercial networks.

The final piece of research is 'Research and design of the distributed intrusion detection system based on Snort' by Zhao Kai [25]. In the paper discusses the ability to use Snort as a centralized network intrusion detection system to protect a campus network by introducing a framework.

False Alarms

As of all detection systems false alarms can occur which could be triggered at any time giving an incorrect state of panic. The cause of a false alarm can be anything from faulty equipment to improper installation therefore procedures must be put in place to prevent this. In a networking environment the reliability of detection systems (Intrusion Detection and Intrusion Prevention) is vital for the protection of equipment and data as the number of network breaches is increasing.

The first paper which will be looked concerning false alarms is 'A robust and fault tolerant distributed intrusion detection system' written by Jaydip Sen from Innovation Lab [22]. The paper starts off discussing that it is 'impossible to predict and identify all vulnerabilities of a network' [22]. This a key point which has been highlight constantly throughout this literature review. Intrusion detection/prevention alone cannot protect a network. J.Sen then continues to say 'IDS needs to be accurate, adaptive and extensible' [22]. With complexity of today's networks and increasing security threats from vulnerable websites, zero-day attacks and increase breaches in data keeping up attackers is becoming increasly difficult. This paper highlights the difficulty deploying IDS technologies because of high rate of false alarms. It is important to test IDS devices before implementing them on a live network to prevent network downtime and disruption to its users.

The first paper in this section looked at the high failure rate of Intrusion Detection systems producing false alarms. The next paper 'False alarm minimization technqiues in signature

based intrusion detection systems' by Neminath Hubbailli and Vinoth Suryannarayanan [23]. In the paper it talks about how in large networking environment it is difficult to manage all alerts especially low level ones. This paper suggests the use of Security and Information Event Management tools (SIEM) to manage these low level events and categorize them. The paper also looks at other methods of reducing false alarms but looking at the pros and cons. The first method in reducing false alarms is introducing state full signatures on a IDS. The pros of this is that if an attack which spans through multiple packets and sessions which allows the IDS to detect multiple stages of attack. Stateful signatures along managing software gives time stamps on what attacker targeted and exploited in the network. This allows disaster recovery teams to piece together the different stages of the attack and reduce the amount of damage by introducing proactive methods. The cons of implementing stateful signatures is that it adds processing overhead. Along with the network speed, analyzing traffic will become increasingly difficult to keep up and log potential attacks. The last downside is that remembering the 'state information' [23] is critical for attacks but IDS does not provide a window to remember them. This paper introduces the pitfalls of configuring IDS technologies incorrectly and not being able log attacks which happen. To combat this the paper recommends using a form SIEM system to log attacks more efficiently and produce greater management tools for the IDS.

Conclusion

This section has looked at the research surrounding the area of cyber security and Intrusion Detection systems. Academic research has been critically evaluated and compared to each other to gain an understanding of the cyber security field and provide the basis for this project. From this research Snort will be used instead of Cisco IPS and rules will be written for test its ability to detect attacks from NMAP, HYDRA and DDoS attacks. The Symantec Security Report has highlighted the key security issues of 2015 which has shaped the way this project is undertaken. .

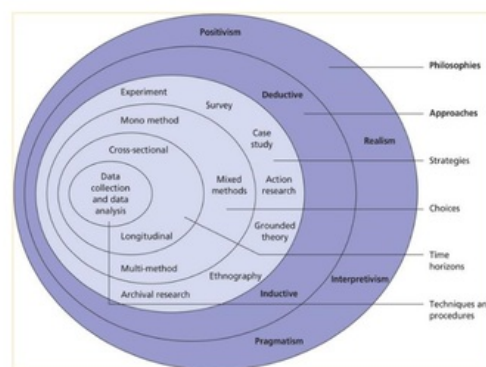
Methodology

This chapter will discuss the research design, philosophy, research methodology and research methods for data collection and how it relates to the aims and objectives of the research. The aim of this chapter is to justify the research design in relation to the objectives and how the research design involves as learning and understanding develops.

The research process of this project will be based on an inductive approach where the aim is to build on the theory which involves collecting data, analysing it and then developing the theory. In this case, a comparison of both Intrusion Detection, Prevention systems will prove that IPS is a more valuable and useful platform. Building on top of this theory, the collected data will prove how much better IPS is. The outcome of this is that more knowledge of this subject will result in a better understanding of the two technologies.

Before starting, a clear philosophy needs to be decided on. Below we can see the analysis of the 'Research Onion' which introduces different philosophies, approaches, strategies, time horizons and techniques/procedures.

Below is the 'Research Onion': [Fig.9]



This research project is an experiment which requires to be conducted in a professional and scientific manner. A suitable philosophy that fits this project is positivism. Positivism suggest that there are single true answers to questions, which can be found by carefully applying scientific procedures. By taking this philosophical approach it will provide a high measure of validity and reliability of results which will result in more successful project as well as creditable. On the other hand a different type of philosophy is phenomenography which is

more human and dynamic. Phenomenographical approach tends to produce more qualitative data which is open to interpretation which produces wide spectrum of data as most people think differently to each other. This provides a good contrast because it can build two arguments for and against each other and may provoke some sort of passion or strong feelings for a subject. To get the best results this theory needs to be deployed in a natural environment to get more real and personalised answers.

Once my project commences a selection of electronic equipment will be used. To collect data from the experiment a series of appropriate tests will be devised to record such data. To do this a series of software and tools are needed to monitor the test bed for organised attacks against the network. As this project is not for public opinion and findings are not open for interpretation qualitative research is not really applicable. Instead quantitative research is a more viable option as it produces statistics and facts known as hard data. The main way of collecting information and data in a scenario like this is through observations and analytical software. This will provide a high degree of validity and reinforcement on findings.

The time horizon used will be longitudinal as the experiment will be conducted over 3 weeks a substantial period of time. As the experiment develops the complexity it will increase by adding new equipment and configurations thus producing different outcomes and results. It is important that a comparison is made before and after introducing an Intrusion Detection system to evaluate the effectiveness. The study population and sample for the project will be fellow students who currently are considered Users of the network. Students will come from the Faculty of Applied Design and Engineering.

After the network has been built and Intrusion Detection is implemented, tests with a couple of users will provide real world data. The aim is to attack them while they are doing set tasks. Once these tests are completed, a series of interviews with the users to talk about any noticeable performance decreases or anything they didn't expect. The answers received back can determine if the Network Intrusion Detection system has worked and if it provided suitable warning for the user. The answers can also help with recommendations for further tests. These interactions will come as a form of structured interview with the User answering a number of set questions. After this the User will be observed carrying out set tasks to monitor a human response once an attack takes place.

Design and Implementation

Introduction

In this section of the project will be split into two parts design and implementation. The design section will state and list the finalised requirements of the project and justify why the experiment used different processes over another. In particular examine the hardware and software used and why it was better than other methods and give reasons for this. The overall aim of this section is to justify why certain process were more beneficial over others in getting the accurate and consistent results and backing this up by referencing back to the literature review section to add weight to the argument.. In second part of this section the Implementation will document the process of setting up the experiment via screen shots and the outcome of each attack. The implementation will be split up into two parts, firstly the examination of SNORT intrusion detection rules and secondly SNORT intrusion prevention rules. At the end of this section it will discuss any problems with the experiment and how they were overcome.

Design

As explained in the introduction this section will talk about the different design decisions which effected the type of software and hardware choices as well as the type of network attacks undertaken. Before discussing these it is important decisions it is important to reference back to the very first chapter of this project where aims and objectives were first listed. Below is original statement of overall aim of this project:

*The overall aim of this project is to assess the effectiveness of Intrusion
Detection/Prevention Technologies and how they can be tuned to be more*

effective in detecting and preventing attacks. The secondary aim is to deliver a detailed guide on different types of network intrusion and detection systems used today in organisations with important emphasis on following industry standard practises by looking at placement, configuration and positives and negatives of each technologies. The outcome of this project is to gain industry knowledge on network intrusion and on how to secure a network using appropriate devices and standards.

As stated above the overall aim is to accurately test the effectiveness of Intrusion Detection/Prevention technologies and how they can be tuned to be more effective. From this overall a requirement specification can be created to make sure the project is successful in achieve its overall aim. Below is the requirement specification created from the aim:

Requirement Specification

Below is a list requirement to meet the overall objectives of the assignment:

- A topology must be created to resemble key and mission critical services either physical or virtualised.
- Some form of firewall must be setup to block any unnecessary traffic to prevent outside alerts effecting the experiments
- Intrusion Detection/Prevention system must be implemented at key areas of the topology in both inline and procoious modes alert or drop packets from the experiments carried out.
- Rules from Intrusion Detection/Prevention systems must be configurable to allow custom rule set from specific attacks.
- Alerts generated must be logged and sort inform of a text document to examination later.

From each of these points sections will be created to justify choosing certain software and hardware over others and how it will benefit the success of the project.

Topology

In this section of the design of the topology will be discussed and different processes will be compared to each other. At the start of the project a original design was created using Cisco routers, switches, firewall and Cisco IP/DS. The idea was to create to separate topologies on with intrusion detection enabled on one while other was configured towards network prevention. Below is a diagram of both topologies:

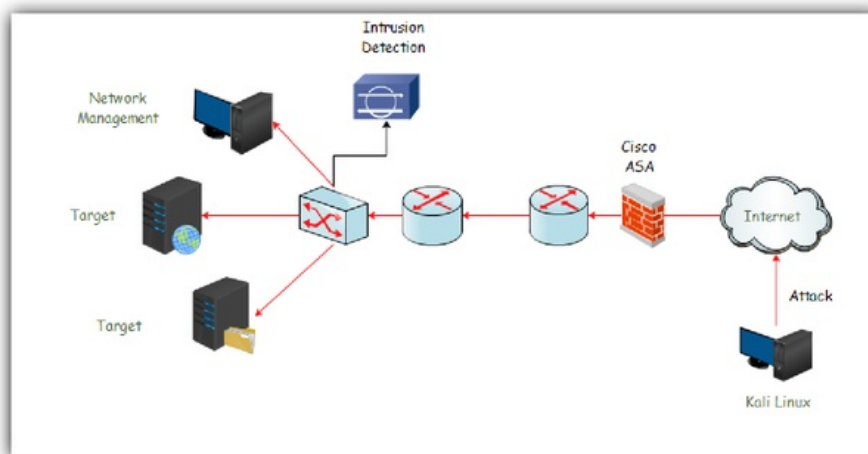


Figure 10 Above is network detection topology using Cisco devices

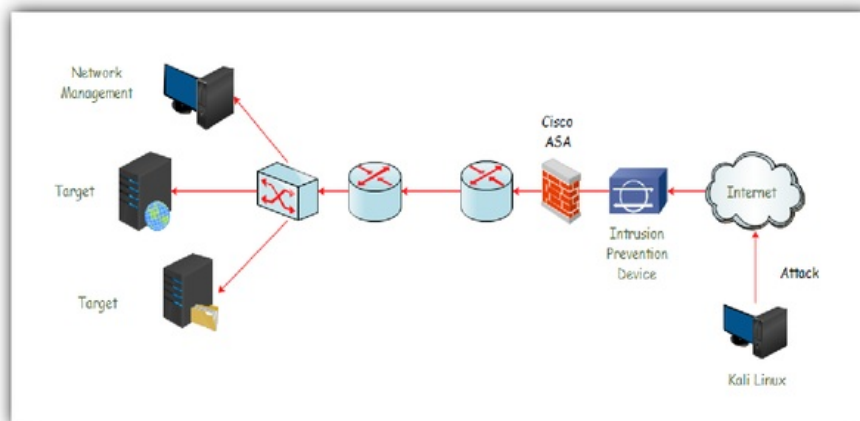


Figure 11: Above is a network prevention topology using Cisco devices:

Why physical setup was not feasible

As you can see both of these topologies use 2 routers a switch, Cisco ASA, Cisco IP/IDs , 2 target machines, 1 pc for managing the IP/DS and pc to run kali Linux penetrating software. At the start this was going to be configured in the security room with physically hardware which is fine and accomplishable. The problem was integrating Cisco IP/Ds either as a physical device or virtually. To implement appliance physically the device would be to be purchased which has its own set of problems. First problem is that device itself is expensive, Cisco release of Next Generation Firewalls with integrated IPS modules being release as a product refresh and being pushed as their flag ship Intrusion Detection/Prevention System which aimed at commercial market and far beyond the reach for this project. This has an impact on legacy series of Cisco IP/Ds products as they were being discontinued thus resulting the loss in support for patches and customer service which is an important factor because this project aims to provide detail on these devices for organisations and business. Another reason why a physical IP/Ds device wasn't feasible was the licence and software required to keep rules and signatures up to date. Legacy IP/Ds devices were being sold with

expired licences thus not having the up to date rule set to combat attacks. It must be pointed out that even without a licence Cisco IP/Ds device is still legal to use but just will not have software support. The next reason why is a physical device was not feasible is the require software called Cisco IPS Manager express. This software was not required to operate the device but made it easier because it interfaced with the sensor to provide a Graphical User Interface. Like with all network management software Cisco IME has bundled a lot of tools to make configuring and monitoring multiple sensors easier which is great for real time monitoring and representing data in an easy format with ability to create graphs. To obtain this software a Cisco Account is needed to purchase the full package or a demo was also available thus making the software unexessable. Below is a screen shot of Cisco IME screen:



Figure 12: Cisco IME

The final point why a physical Cisco IP/Ds wasn't feasible was the rack space needed to store the appliance securely was not available due to other routers, switches and ASAs for other students. Moving these devices would be unfair on other students who wish to study on the equipment and potential cause disruption to classes.

Virtual Setup with Cisco IP/Ds Device

Now we have established that a physical implementation of a Cisco IP/Ds is not possible due to reasons above we move onto implementing a virtualised version of sensor. From research carried out there are two ways to do this. One using GNS3 software used by network professionals to practice simulating networks and two using Virtual box OVA file to run as a virtual machine. From research there were available ISO files split into 'disks' which could be emulated via QEMU using GNS3. Below is Cisco IP/Ds setup in GNS3:

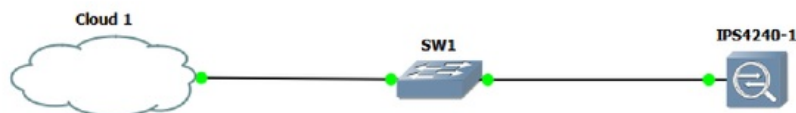


Figure 13: Cisco IPS in GNS3

The topology above consists of a 'Cloud' which is a Microsoft Loopback Interface, a Switch and the Cisco IPS device. Below is the basic configuration of the IP/Ds:

```
sensor#
sensor# conf t
sensor(config)# service host
sensor(config-hos)# network-settings
sensor(config-hos-net)# host-ip 200.200.200.7/24,200.200.200.1
sensor(config-hos-net)# host
host-ip      host-name
sensor(config-hos-net)# host-name IPS
sensor(config-hos-net)# access-list 200.200.200.0/24
sensor(config-hos-net)# exit
sensor(config-hos)# exit
Apply Changes?[yes]: yes
```

Figure 14: Cisco IPS configuration

From this screen shot above the configuration named, gave an IP address and configure basic access control list for the PC connected with loopback interface to access. In the previous chapter, it was highlighted that Cisco ISE is recommended to interface with the sensor but was unattainable due to it being purchasable but also came with a demo version. From additional research a demo version was found and installed to test the feasibility of using a virtualised sensor with GNS3. Interfacing with the sensor via Cisco ISE was somewhat successful with being able to configure and modify rules but the licence on the sensor was over 3 years out of date so therefore unusable to test against attacks which have been updated constantly. Also sometimes there were issues between the switch and the sensor as they drop packets and cause glitches. From further research, GNS3 currently cannot simulate layer 2 device and therefore emulates a lot simple hub instead. This caused major issues because one of the features of the Cisco IP/Ds is to monitor VLAN traffic through inline mode. Not being able to configure this on a standard hub was the final nail in the coffin for trying to emulate a Cisco IP/Ds device. There was stability issues with GNS3 crashing and becoming unresponsive and not at all good for trying to create a guide for business.

SNORT

From the failure in trying to emulate a Cisco IP/Ds device an open source Intrusion Detection/Prevention system a more viable option. From research SNORT was the best option as it is widely supported and has a large community as well as a lot of research behind it. SNORT is versatile as it can be installed Windows and Linux based systems and is supported with other open source software like BASE, Snortby, Squel which are all open source GUI interface to manage the sensor. SNORT can also be implemented on a 'SNORT box' a device which solely runs SNORT like a Cisco IP/Ds. Below is an example of a SNORT topology:

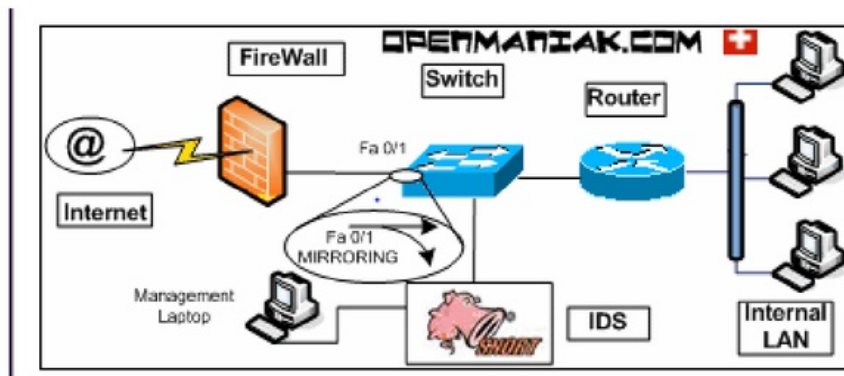


Figure 15: Above is an example of a SNORT box sniffing packets

The advantage of SNORT it can be implemented in virtual environment and integrated with Kali Linux which is the gold standard for penetration testing. Another advantage of SNORT is that it can run PCAP files. PCAP files are pre-recorded traffic which can be used maliciously to generate alerts used to test ruleset. This is used before adding SNORT to a live network to test rule set for any vulnerabilities. Another advantage of SNORT is that the rules are easily accessible the SNORT rules file and easily made changes to on the fly. To support the configuration of the SNORT rules there is a whole documentation provided by SNORT to create reliable rules. An alternative to this is to download latest rules created by the community which provides good protection. For higher protection, there are also subscription for paid services for more advanced rule set.

A recommendation of SNORT package is Security Onion.

Firewall

Stop any inference from outside sources a firewall is needed to block any unnecessary traffic. This was finalised in the physically topology using a Cisco ASA. But with the dropping of trying to emulate a Cisco IP/Ds the need for a new solution. The new solution had to be able to be virtualised and compatible with SNORT. From research, Pfsense again an open source

firewall was discovered. Like SNORT PfSense can run a virtual machine and has DHCP capabilities for other virtual machines on the network. PfSense has GUI interface which can be accessed from a server to manage the firewall and install extra packages. PfSense has built-in package for SNORT with its own GUI interface where it can be activated. Ideally it is best practice to manage SNORT on a standalone machine.

Testing

As talked about earlier Kali Linux is the gold standard for penetration testing various sort of vulnerabilities in a network. Kali Linux has NMAP network scanning, Metasploit Framework, DDoS and other tools to test ID/PS. It also responsible resource heavy running as a virtual machine with recommended requirement of 2GB RAM being assigned to it. Kali Linux also has a lot of unnecessary applications for this project. Alternative to this is create custom testing virtual machine using cherry picked tools from Kali. The result of this is that it will be less resource intensive. The third and final option is to run PCAPs. PCAPs are re-recorded traffic with malicious content in them which is guaranteed to create alert.

Final Test Bed

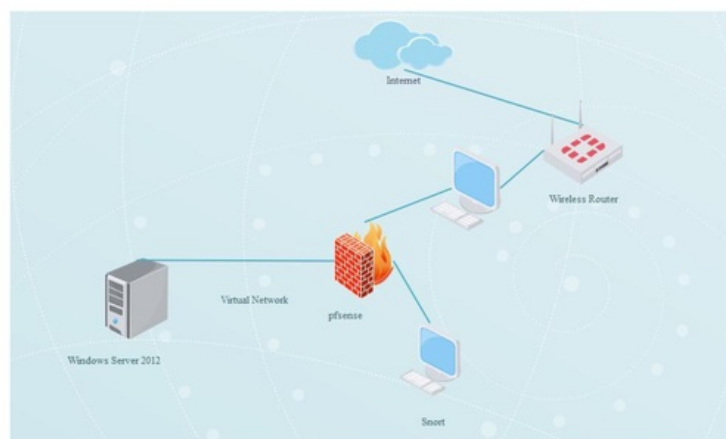


Figure 16: Final test bed

Implementation

This section of the report will look at the practical side of the project will look at the custom ruleset made for combatting DDoS, Hydra, Nmap attacks. Before we look into the ruleset we must look at the experiment test bed in detail. Below is a diagram of the topology:

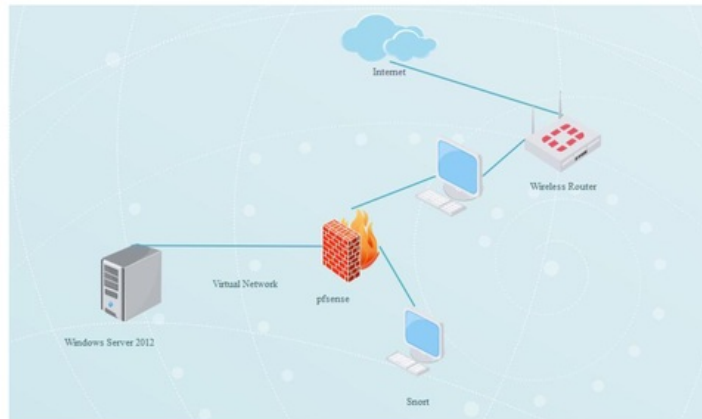


Figure 17: Final Test Bed

The topology is made up from the following:

- PfSense
- Windows Server 2012
- Snort

Explanation of Test bed

Due to problems discussed the 'Design Section' of this report, Cisco IPS was not able to be emulated through GNS3 or virtual OVA format due to crashing and stability issues with layer 2 devices. Instead, the experiment will be virtualized and instead of using Cisco IPS Snort an open source intrusion detection will be used instead. Snort will be accompanied by pfSense which is an open source firewall to act as a DHCP server for clients. Snort is able to be installed as a package on pfSense but to have more control over Snort rules it has been installed on a Windows 10 machine. Below is screen shot of snort:

```

C:\Users\Robert>snort
Running in packet dump mode

--- Initializing Snort ---
Initializing Output Plugins!
pcap DAQ configured to passive.
The DAQ version does not support reload.
Acquiring network traffic from "\Device\NPF_{A3B042CC-0700-47CB-B9C2-81B77097CCF9}".
Decoding Ethernet

--- Initialization Complete ---

o"~
...
    -*> Snort! <*-
    Version 2.9.8.2-WIN32 GRE (Build 335)
    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
    Copyright (C) 2014-2015 Cisco and/or its affiliates. All rights reserved.
    Copyright (C) 1998-2013 Sourcefire, Inc., et al.
    Using PCRE version: 8.10 2010-06-25
    Using ZLIB version: 1.2.3

```

Figure 18: Snort

Example of false alarms

Snort sniffs packets from the virtual Ethernet cable connected to virtual test bed, false alarms would generate if the user would open up the browser for example, below is a screenshots:

```

=====
Run time for packet processing was 12.152000 seconds
Snort processed 40 packets.
Snort ran for 0 days 0 hours 0 minutes 12 seconds
  Pkts/sec:          3
=====
Packet I/O Totals:
  Received:          41
  Analyzed:           40 ( 97.561%)
  Dropped:            0 (  0.000%)
  Filtered:           0 (  0.000%)
  Outstanding:        1 (  2.439%)
  Injected:           0
=====
Breakdown by protocol (includes rebuilt packets):
  Eth:                40 (100.000%)
  VLAN:                0 (  0.000%)
  IP4:                 37 ( 92.500%)
  Frag:                0 (  0.000%)
  ICMP:                0 (  0.000%)
  UDP:                14 ( 35.000%)
  TCP:                 23 ( 57.500%)
  IP6:                 2 (  5.000%)
  IP6 Ext:             2 (  5.000%)
  IP6 Opts:            0 (  0.000%)
  Frag6:               0 (  0.000%)
  ICMP6:               0 (  0.000%)
  UDP6:                2 (  5.000%)

```

Figure 19: False Alarms

```

GRE IP4:      0 ( 0.000%)
GRE IP6:      0 ( 0.000%)
GRE IP6 Ext:  0 ( 0.000%)
GRE PPTP:     0 ( 0.000%)
GRE ARP:      0 ( 0.000%)
GRE IPX:      0 ( 0.000%)
GRE Loop:     0 ( 0.000%)
MPLS:         0 ( 0.000%)
  ARP:        1 ( 2.500%)
  IPX:        0 ( 0.000%)
Eth Loop:     0 ( 0.000%)
Eth Disc:     0 ( 0.000%)
IP4 Disc:     0 ( 0.000%)
IP6 Disc:     0 ( 0.000%)
TCP Disc:     0 ( 0.000%)
UDP Disc:     0 ( 0.000%)
ICMP Disc:    0 ( 0.000%)
All Discard:  0 ( 0.000%)
  Other:      0 ( 0.000%)
Bad Chk Sum:  12 ( 30.000%)
Bad TTL:      0 ( 0.000%)
SS G 1:       0 ( 0.000%)
SS G 2:       0 ( 0.000%)
Total:        40

```

Figure 20: False Alarms

As you can see SNORT ran for 12 seconds and analyzed 40 different packets, we want to setup a test area where Snort can generate alerts relating to the traffic sent by PCAP.

Pfsense Managing the Firewall

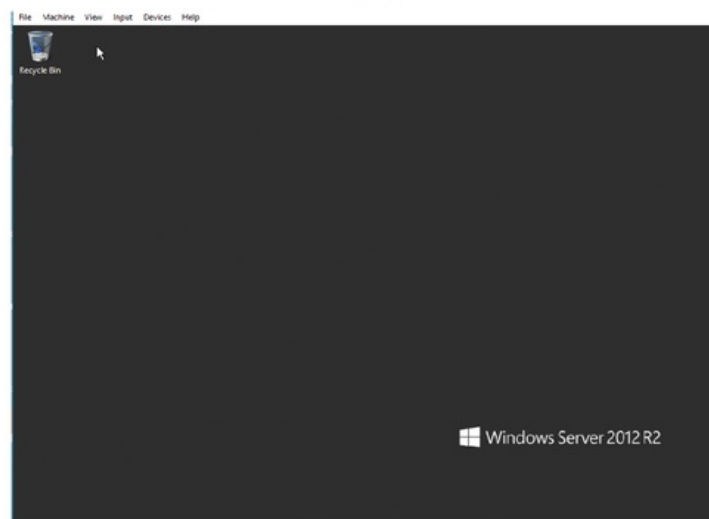


Figure 21: Windows 2012

Below Pfsense with ip address of 10.0.0.1

```

File Machine View Input Devices Help
Starting CRON... done.
pfSense (pfSense) 2.3-RELEASE amd64 Mon Apr 11 18:10:34 CDT 2016
Bootup complete

FreeBSD/amd64 (pfSense.localdomain) (ttyv0)

*** Welcome to pfSense 2.3-RELEASE-pfSense (amd64) on pfSense ***

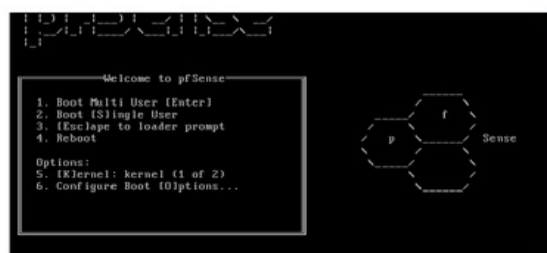
WAN (wan)      -> em0      -> v4/DHCP4: 192.168.0.22/24
                v6/DHCP6: fd6a:cc14:87c6:0:a00:27ff:fed2:bcd/64
LAN (lan)      -> em1      -> v4: 10.0.0.1/24

0) Logout (SSH only)          9) pfTop
1) Assign Interfaces          10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) pfSense Developer Shell
4) Reset to factory defaults  13) Update from console
5) Reboot system              14) Enable Secure Shell (sshd)
6) Halt system                 15) Restore recent configuration
7) Ping host                   16) Restart PHP-FPM
8) Shell

Enter an option:

```

Figure 22: Pfsense



23: Pfsense

Above is pfsense.

Hydra Attack

The first attack to test Snort is doing to be a Hydra Attack. A hydra attack is a brute force attack which targets passwords. In this situation hydra attack could be attacking server to gain FTP access. Below is the Snort rule to detect this:

```

alert tcp any 21 -> any any (msg:"FTP Bad login"; content:"530 User "; nocase;
flow:from_server,established; sid:491; rev:5;)

```

As you can see above, the signature written name 'FTP Bad Login' alerts any tcp traffic which tries to use port 21 from the server with the signature of 491.

Below is the wireshark pcap, with details of Hydra Attack on Microsoft FTP:

Figure 26: Hydra Attack

As you can see above, the hydra_ftp.pcap file has been loaded in along with rules.rules file which should detect it, below is command to run the hydra_ftp.pcap file along with rules.rules:

```
C:\mysnort>snort -r hydra_ftp.pcap -c rules.rules
Running in IDS mode
```

Figure 27: Running Attack with rules

Snort Output

As you can see below there were 307 packets with 100% were analyzed

```
Commencing packet processing (pid=2136)
=====
Run time for packet processing was 0.2000 seconds
Snort processed 307 packets.
Snort ran for 0 days 0 hours 0 minutes 0 seconds
  Pkts/sec:          307
=====
Packet I/O Totals:
  Received:          307
  Analyzed:          307 (100.000%)
  Dropped:            0 ( 0.000%)
  Filtered:           0 ( 0.000%)
  Outstanding:        0 ( 0.000%)
  Injected:           0
```

Figure 28: Hydra Output

From those 307 packet 41 packets were alerted and logged which contained Hydra_FTP data

```
=====
Action Stats:
  Alerts:          41 ( 13.312%)
  Logged:          41 ( 13.312%)
  Passed:           0 (  0.000%)
Limits:
  Match:           0
  Queue:           0
  Log:             0
  Event:           0
  Alert:           0
Verdicts:
  Allow:          307 (100.000%)
  Block:           0 (  0.000%)
  Replace:         0 (  0.000%)
  Whitelist:       0 (  0.000%)
  Blacklist:       0 (  0.000%)
  Ignore:          0 (  0.000%)
  (null):          0 (  0.000%)
=====
```

Figure 29: Hydra Output

Below are the Snort stream stats:

```

=====
Stream statistics:
  Total sessions: 16
  TCP sessions: 16
  UDP sessions: 0
  ICMP sessions: 0
  IP sessions: 0
  TCP Prunes: 0
  UDP Prunes: 0
  ICMP Prunes: 0
  IP Prunes: 0
TCP StreamTrackers Created: 16
TCP StreamTrackers Deleted: 16
  TCP Timeouts: 0
  TCP Overlaps: 0
  TCP Segments Queued: 84
  TCP Segments Released: 84
  TCP Rebuilt Packets: 16
  TCP Segments Used: 84
  TCP Discards: 0
  TCP Gaps: 15
  UDP Sessions Created: 0
  UDP Sessions Deleted: 0
  UDP Timeouts: 0
  UDP Discards: 0
  Events: 0
  Internal Events: 0
  TCP Port Filter
    Filtered: 0
    Inspected: 0
    Tracked: 307
  UDP Port Filter
    Filtered: 0
    Inspected: 0
    Tracked: 0
=====

```

Figure 30: Hydra Output

Snort Log files:

```

Directory of C:\mysnort\log
20/04/2016  20:51    <DIR>          .
20/04/2016  20:51    <DIR>          ..
20/04/2016  20:51                23,116 alert.ids
20/04/2016  19:46                 0 portscan.log
20/04/2016  19:46                4,698 snort.log.1461177994
20/04/2016  20:51                4,698 snort.log.1461181900
               4 File(s)          32,512 bytes
               2 Dir(s)  93,071,659,008 bytes free

```

Figure 31: Alerts

As you can see the Snort has logged the alerts
To view the sorted alerts type 'notepad alert.ids'

```
C:\mysnort\log>notepad alert.ids
```

Figure 32: alerts

A new window pops up with all the logs

```

alerLids - Notepad
File Edit Format View Help
[**] [1:491:5] FTP Bad login [**]
[Priority: 0]
01/04-11:19:34.252082 192.168.75.132:21 -> 192.168.75.1:18163
TCP TTL:128 TOS:0x0 ID:1743 IpLen:20 DgmLen:82 DF
***AP*** Seq: 0xFACD9194 Ack: 0x4E2AB853 Win: 0xFADA Tcplen: 32
TCP Options (3) => NOP NOP TS: 13957 7457681

[**] [1:491:5] FTP Bad login [**]
[Priority: 0]
01/04-11:19:34.255330 192.168.75.132:21 -> 192.168.75.1:18164
TCP TTL:128 TOS:0x0 ID:1744 IpLen:20 DgmLen:82 DF
***AP*** Seq: 0x5F3D1C86 Ack: 0x94A27AAC Win: 0xFADA Tcplen: 32
TCP Options (3) => NOP NOP TS: 13957 7457681

[**] [1:491:5] FTP Bad login [**]
[Priority: 0]
01/04-11:19:34.256494 192.168.75.132:21 -> 192.168.75.1:18162
TCP TTL:128 TOS:0x0 ID:1745 IpLen:20 DgmLen:83 DF
***AP*** Seq: 0x76C91E0B Ack: 0x68B77C72 Win: 0xFAD5 Tcplen: 32
TCP Options (3) => NOP NOP TS: 13958 7457681

```

Figure 32: Notepad log
DDoS

In the next attack we will look at Denial of Service, in this case denial of service attack can be aimed at the Windows 2012 server to stop access for clients. Below is the rule for detecting DDoS attacks:

```

alert tcp any any -> any 80 (msg:"DOS flood denial of service attempt";flow:to_server;
\
detection_filter:track by_dst, count 60, seconds 60; \
sid:25101; rev:1);

```

As you can see above the DoS attack is written for a FYN flood, with alert all TCP traffic with a sid of 25101.

Below are the details a flood attack in Wireshark:

1	0.000000	192.168.75.132	192.168.75.1	TCP	58 1140 → 8 [190] Seq=61001121 Len=0
2	0.000012	192.168.75.132	192.168.75.1	TCP	58 1140 → 8 [190] Seq=61001121 Len=0
3	0.000028	192.168.75.132	192.168.75.1	TCP	58 1140 → 8 [190] Seq=61001121 Len=0
4	0.000044	192.168.75.132	192.168.75.1	TCP	58 1140 → 8 [190] Seq=61001121 Len=0
5	0.000060	192.168.75.132	192.168.75.1	TCP	58 1140 → 8 [190] Seq=61001121 Len=0
6	0.000076	192.168.75.132	192.168.75.1	TCP	58 1140 → 8 [190] Seq=61001121 Len=0
7	0.000092	192.168.75.132	192.168.75.1	TCP	58 1140 → 8 [190] Seq=61001121 Len=0
8	0.000108	192.168.75.132	192.168.75.1	TCP	58 1140 → 8 [190] Seq=61001121 Len=0
9	0.000124	192.168.75.132	192.168.75.1	TCP	58 1140 → 8 [190] Seq=61001121 Len=0
10	0.000140	192.168.75.132	192.168.75.1	TCP	58 1140 → 8 [190] Seq=61001121 Len=0
11	0.000156	192.168.75.132	192.168.75.1	TCP	58 1140 → 8 [190] Seq=61001121 Len=0
12	0.000172	192.168.75.132	192.168.75.1	TCP	58 1140 → 8 [190] Seq=61001121 Len=0
13	0.000188	192.168.75.132	192.168.75.1	TCP	58 1140 → 8 [190] Seq=61001121 Len=0
14	0.000204	192.168.75.132	192.168.75.1	TCP	58 1140 → 8 [190] Seq=61001121 Len=0
15	0.000220	192.168.75.132	192.168.75.1	TCP	58 1140 → 8 [190] Seq=61001121 Len=0
16	0.000236	192.168.75.132	192.168.75.1	TCP	58 1140 → 8 [190] Seq=61001121 Len=0
17	0.000252	192.168.75.132	192.168.75.1	TCP	58 1140 → 8 [190] Seq=61001121 Len=0
18	0.000268	192.168.75.132	192.168.75.1	TCP	58 1140 → 8 [190] Seq=61001121 Len=0
19	0.000284	192.168.75.132	192.168.75.1	TCP	58 1140 → 8 [190] Seq=61001121 Len=0
20	0.000300	192.168.75.132	192.168.75.1	TCP	58 1140 → 8 [190] Seq=61001121 Len=0
21	0.000316	192.168.75.132	192.168.75.1	TCP	58 1140 → 8 [190] Seq=61001121 Len=0
22	0.000332	192.168.75.132	192.168.75.1	TCP	58 1140 → 8 [190] Seq=61001121 Len=0
23	0.000348	192.168.75.132	192.168.75.1	TCP	58 1140 → 8 [190] Seq=61001121 Len=0
24	0.000364	192.168.75.132	192.168.75.1	TCP	58 1140 → 8 [190] Seq=61001121 Len=0
25	0.000380	192.168.75.132	192.168.75.1	TCP	58 1140 → 8 [190] Seq=61001121 Len=0
26	0.000396	192.168.75.132	192.168.75.1	TCP	58 1140 → 8 [190] Seq=61001121 Len=0
27	0.000412	192.168.75.132	192.168.75.1	TCP	58 1140 → 8 [190] Seq=61001121 Len=0
28	0.000428	192.168.75.132	192.168.75.1	TCP	58 1140 → 8 [190] Seq=61001121 Len=0
29	0.000444	192.168.75.132	192.168.75.1	TCP	58 1140 → 8 [190] Seq=61001121 Len=0
30	0.000460	192.168.75.132	192.168.75.1	TCP	58 1140 → 8 [190] Seq=61001121 Len=0

Figure 33: Wireshark DDoS

Testing Snort from a DDoS attack

```
Volume in drive C has no label.  
Volume Serial Number is 8C33-36E8  
  
Directory of C:\mysnort  
  
20/04/2016  22:47    <DIR>          .  
20/04/2016  22:47    <DIR>          ..  
25/09/2012  20:27                2,088 hping_syn.pcap  
20/04/2016  22:48    <DIR>          log  
08/01/2014  10:29                3,103 rules.rules  
          2 File(s)              5,191 bytes  
          3 Dir(s)  92,744,712,192 bytes free
```

Figure 34: DDoS

Start DDoS

```
3 Dir(s) 92,744,712,192 bytes free
C:\mysnort>snort -r hping_syn.pcap -c rules.rules
```

Figure 35: DDoS rules

Snort Output

```

o*) Snort! <+
Version 2.9.8.2-WIN32 GRE (Build 335)
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2015 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 SourceFire, Inc., et al.
Using PCRE version: 8.10 2010-06-25
Using ZLIB version: 1.2.3

Commencing packet processing (pid=8684)
-----
Run time for packet processing was 0.0 seconds
Snort processed 28 packets,
Snort ran for 0 days 0 hours 0 minutes 0 seconds
Pkts/sec:      28
-----
Packet I/O Totals:
Received:       28
Analyzed:       28 (100.000%)
Dropped:        0 ( 0.000%)
Filtered:        0 ( 0.000%)
Outstanding:     0 ( 0.000%)
Infected:        0
-----
Breakdown by protocol (includes rebuilt packets):
Eth:            28 (100.000%)
VLAN:           0 ( 0.000%)
IP4:            26 ( 92.857%)
Frag:           0 ( 0.000%)
ICMP:           0 ( 0.000%)
UDP:            1 ( 3.571%)
TCP:           25 ( 89.286%)
IP6:            0 ( 0.000%)
IP6 Ext:        0 ( 0.000%)
IP6 Opts:       0 ( 0.000%)
Frag6:          0 ( 0.000%)
ICMP6:         0 ( 0.000%)
UDP6:           0 ( 0.000%)
TCP6:          0 ( 0.000%)
Teredo:         0 ( 0.000%)
ICMP-IP:       0 ( 0.000%)
EAPOL:         0 ( 0.000%)
TP4/IP4:       0 ( 0.000%)
TP4/IP6:       0 ( 0.000%)
IP6/IP4:       0 ( 0.000%)
IP6/IP6:       0 ( 0.000%)
GRE:           0 ( 0.000%)
GRE Eth:       0 ( 0.000%)
GRE VLAN:     0 ( 0.000%)
GRE IP4:      0 ( 0.000%)
GRE IP6:      0 ( 0.000%)
GRE IP6 Ext:  0 ( 0.000%)
GRE PPtP:     0 ( 0.000%)
GRE ARP:      0 ( 0.000%)

```

Figure 36: DDos

```

=====
Stream statistics:
    Total sessions: 26
      TCP sessions: 25
      UDP sessions: 1
      ICMP sessions: 0
      IP sessions: 0
      TCP Prunes: 0
      UDP Prunes: 0
      ICMP Prunes: 0
      IP Prunes: 0
TCP StreamTrackers Created: 25
TCP StreamTrackers Deleted: 25
      TCP Timeouts: 0
      TCP Overlaps: 0
      TCP Segments Queued: 0
      TCP Segments Released: 0
      TCP Rebuilt Packets: 0
      TCP Segments Used: 0
      TCP Discards: 0
      TCP Gaps: 0
      UDP Sessions Created: 1
      UDP Sessions Deleted: 1
      UDP Timeouts: 0
      UDP Discards: 0
      Events: 25
      Internal Events: 0
      TCP Port Filter
        Filtered: 0
        Inspected: 0
        Tracked: 25
      UDP Port Filter
        Filtered: 0
        Inspected: 0
        Tracked: 1
=====

```

Figure 37: DDoS

Nmap

In the next attack Snort will attempt to detect a NMAP attack, NMAP is a port scanner software which shows which ports are open. Below is the rule for NMAP port scan:

```

preprocessor sfportscan:\
    proto { all } \
    scan_type { all } \
    sense_level { high } \
    logfile { portscan.log }

```

As you can see above this is the ruleset for NMAP port scan, with the threat set to high.

Unlike hydra attack the alert is logged in portscan.log. Below is the wireshark output:

1	0.000000	192.168.75.132	192.168.75.132	80	77 Standard query request from 192.168.75.132 to 192.168.75.132
2	0.000000	192.168.75.132	192.168.75.132	80	80 Standard query response from 192.168.75.132 to 192.168.75.132
3	0.000000	192.168.75.132	192.168.75.132	2209	74 DNS [44] request 192.168.75.132 to 192.168.75.132
4	0.000000	192.168.75.132	192.168.75.132	2209	74 DNS [44] response 192.168.75.132 to 192.168.75.132
5	0.000000	192.168.75.132	192.168.75.132	70	82 DNS [44] request 192.168.75.132 to 192.168.75.132
6	0.000000	192.168.75.132	192.168.75.132	70	82 DNS [44] response 192.168.75.132 to 192.168.75.132
7	0.000000	192.168.75.132	192.168.75.132	70	82 DNS [44] request 192.168.75.132 to 192.168.75.132
8	0.000000	192.168.75.132	192.168.75.132	70	82 DNS [44] response 192.168.75.132 to 192.168.75.132
9	0.000000	192.168.75.132	192.168.75.132	70	82 DNS [44] request 192.168.75.132 to 192.168.75.132
10	0.000000	192.168.75.132	192.168.75.132	70	82 DNS [44] response 192.168.75.132 to 192.168.75.132
11	0.000000	192.168.75.132	192.168.75.132	70	82 DNS [44] request 192.168.75.132 to 192.168.75.132
12	0.000000	192.168.75.132	192.168.75.132	70	82 DNS [44] response 192.168.75.132 to 192.168.75.132
13	0.000000	192.168.75.132	192.168.75.132	70	82 DNS [44] request 192.168.75.132 to 192.168.75.132
14	0.000000	192.168.75.132	192.168.75.132	70	82 DNS [44] response 192.168.75.132 to 192.168.75.132
15	0.000000	192.168.75.132	192.168.75.132	70	82 DNS [44] request 192.168.75.132 to 192.168.75.132
16	0.000000	192.168.75.132	192.168.75.132	70	82 DNS [44] response 192.168.75.132 to 192.168.75.132
17	0.000000	192.168.75.132	192.168.75.132	70	82 DNS [44] request 192.168.75.132 to 192.168.75.132
18	0.000000	192.168.75.132	192.168.75.132	70	82 DNS [44] response 192.168.75.132 to 192.168.75.132
19	0.000000	192.168.75.132	192.168.75.132	70	82 DNS [44] request 192.168.75.132 to 192.168.75.132
20	0.000000	192.168.75.132	192.168.75.132	70	82 DNS [44] response 192.168.75.132 to 192.168.75.132
21	0.000000	192.168.75.132	192.168.75.132	70	82 DNS [44] request 192.168.75.132 to 192.168.75.132
22	0.000000	192.168.75.132	192.168.75.132	70	82 DNS [44] response 192.168.75.132 to 192.168.75.132
23	0.000000	192.168.75.132	192.168.75.132	70	82 DNS [44] request 192.168.75.132 to 192.168.75.132
24	0.000000	192.168.75.132	192.168.75.132	70	82 DNS [44] response 192.168.75.132 to 192.168.75.132
25	0.000000	192.168.75.132	192.168.75.132	70	82 DNS [44] request 192.168.75.132 to 192.168.75.132
26	0.000000	192.168.75.132	192.168.75.132	70	82 DNS [44] response 192.168.75.132 to 192.168.75.132
27	0.000000	192.168.75.132	192.168.75.132	70	82 DNS [44] request 192.168.75.132 to 192.168.75.132
28	0.000000	192.168.75.132	192.168.75.132	70	82 DNS [44] response 192.168.75.132 to 192.168.75.132
29	0.000000	192.168.75.132	192.168.75.132	70	82 DNS [44] request 192.168.75.132 to 192.168.75.132
30	0.000000	192.168.75.132	192.168.75.132	70	82 DNS [44] response 192.168.75.132 to 192.168.75.132
31	0.000000	192.168.75.132	192.168.75.132	70	82 DNS [44] request 192.168.75.132 to 192.168.75.132
32	0.000000	192.168.75.132	192.168.75.132	70	82 DNS [44] response 192.168.75.132 to 192.168.75.132
33	0.000000	192.168.75.132	192.168.75.132	70	82 DNS [44] request 192.168.75.132 to 192.168.75.132
34	0.000000	192.168.75.132	192.168.75.132	70	82 DNS [44] response 192.168.75.132 to 192.168.75.132

Figure 38: Wireshark Nmap
Testing NMAP port scan attack

```

Directory of C:\mysnort

20/04/2016  22:05    <DIR>          .
20/04/2016  22:05    <DIR>          ..
06/01/2014  23:57             27,590  arp_spoof.pcap
25/09/2012  20:27             2,088  hping_syn.pcap
20/04/2016  22:05    <DIR>          log
25/09/2012  20:27             172,921  nmap.pcap
08/01/2014  10:29             3,103  rules.rules
               4 File(s)          205,702 bytes
               3 Dir(s)  92,775,518,208 bytes free

C:\mysnort>

```

Figure 39: Nmap

As you can see above, the nmap.pcap file has been loaded in along with rules.rules file which should detect it, below is command to run the nmap.pcap file along with rules.rules:

```

3 Dir(s)  92,775,518,208 bytes free

C:\mysnort>snort -r nmap.pcap -c rules.rules

```

Figure 40: NMAP rules

Snort Output

```
..-_*~
o"~)~
....

-*> Snort! <*-
Version 2.9.8.2-WIN32 GRE (Build 335)
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2015 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using PCRE version: 8.10 2010-06-25
Using ZLIB version: 1.2.3

Commencing packet processing (pid=2376)
=====
Run time for packet processing was 0.4000 seconds
Snort processed 2270 packets.
Snort ran for 0 days 0 hours 0 minutes 0 seconds
Pkts/sec: 2270
=====
Packet I/O Totals:
Received: 2270
Analyzed: 2270 (100.000%)
Dropped: 0 ( 0.000%)
Filtered: 0 ( 0.000%)
Outstanding: 0 ( 0.000%)
Injected: 0
=====
```

Figure 41: Nmap Outcome

Above 2270 packets were analysed

```
=====
Stream statistics:
Total sessions: 1090
TCP sessions: 1087
UDP sessions: 3
ICMP sessions: 0
IP sessions: 0
TCP Prunes: 0
UDP Prunes: 0
ICMP Prunes: 0
IP Prunes: 0
TCP StreamTrackers Created: 1087
TCP StreamTrackers Deleted: 1087
TCP Timeouts: 0
TCP Overlaps: 0
TCP Segments Queued: 32
TCP Segments Released: 32
TCP Rebuilt Packets: 14
TCP Segments Used: 32
TCP Discards: 0
TCP Gaps: 2
UDP Sessions Created: 3
UDP Sessions Deleted: 3
UDP Timeouts: 0
UDP Discards: 0
Events: 0
Internal Events: 0
TCP Port Filter
Filtered: 0
Inspected: 0
Tracked: 2257
UDP Port Filter
Filtered: 0
Inspected: 0
Tracked: 3
=====
Snort exiting
```

Figure 42: Nmap Outcome

Logged files

```
C:\mysnort>cd log
C:\mysnort\log>dir
Volume in drive C has no label.
Volume Serial Number is 8C33-36E8

Directory of C:\mysnort\log

20/04/2016  22:19    <DIR>          .
20/04/2016  22:19    <DIR>          ..
20/04/2016  22:03                0 alert.ids
20/04/2016  22:19                500 portscan.log
                2 File(s)                500 bytes
                2 Dir(s)  92,773,650,432 bytes free

C:\mysnort\log>notepad portscan.log
C:\mysnort\log>
```

Figure 43: Nmap Logged Files

As you can see above there are no logged files under 'alerts.ids' instead they are under 'portscan.log'

Below is the log file:



Figure 44: Nmap Log

As you can see above the PCAP file was NMAP scanning the ip address range 192.168.75.1 to 192.168.75.132

Conclusion

From the experiments carried out above, testing out Snort against NMAP, DDoS attack and Hydra attack it shows that Snort is capable of detecting the attacks and logging them. The research carried out in literature review said that Snort can log, alert and detect bad traffic. The tests above have proven this. I expected Snort to log all attacks, it did with NMAP and Hydra Scans but not with DDoS. I wasn't expecting this. It could maybe need a special log file. The multiple experiments carried out show that is a correlation of detecting the attacks.

Evaluation

In this section the result of the experiment will be evaluated and compared to the literature and if they met the requirements set in the aims and objectives in the introductory chapter.

Below are both the aims and objectives which are stated:

1. Create a Virtual box network using SNORT, Kali Linux and Windows Server 2012
2. Configure SNORT rules in both IDS/IPS modes to monitor for PING, NMAP and DDoS attacks.
3. Produce a comprehensive comparison between SNORT IDS and Cisco IPS rule set.
4. Analyse the results.

From these objectives the project was mostly successful. A topology was created in virtual box with Snort, Windows Server 2012 but using PCAPs instead of Kali Linux. This was done to get more accurate results and take up less system RAM which could be instead used for Snorts overhead. The attacks carried out against the topology were NMAP, Hydra and DDoS all aimed at disrupting network services. Due to limitations discussed in the design section emulating Cisco IPS was not possible so this way the project failed to meet its target here. The results were analysed in depth using Snort and Logs were produced detailing each attack which met the requirements for the final point.

The overall aim of the project was to gain knowledge on both IPS and IDS systems. Even though IPS device was not reliably emulated, knowledge gained from research and trial and error in using Cisco ISE. Both Snort and Cisco IPS are commercial devices so gaining experience with them is beneficial for someone who wants to enter the network security field. From research in the literature review both IDS and IPS alone are not enough to provide complete security protection to a network. Integrating them with other appliances like firewalls provides a more complete solution as there is no silver bullet to solve all problems. If the project was going to be carried out again it would be more beneficial to use more open source security technologies as they more flexible in being setup and configured. Being more organised is another factor which could of improved this project as well but will be discussed in the conclusion and reflect chapters.

Conclusion and Recommendation

In this section of the report will sum up, conclude and recommend areas of further research. At the start of the project the aims and objectives were summed stated with overall goal was to test and compare both Intrusion Detection Systems and Prevention Systems. At the start of the introductory chapter it introduced the concept of network intrusion and explained it to be an illegal attempt to breach a network and do damage to network devices, infrastructure or to comprise data integrity. To protect data and network devices a security system was needed to provide a deep level of security and there was no 'silver bullet solution' in were there was a device capable of providing the necessary levels of protection. The chapter then introduced the experimental side of the project by deciding to compare Cisco IPS and SNORT by testing each technologies ruleset and signatures. Then the aim and objectives of the project were laid out giving a detail guide on how the practical will be accomplished. The final part of the introductory chapter was the brief break down of all chapters and their contents show the level of procession through the project.

Moving on from the introductory chapter, the literature aimed to review and critically analyses academic research conducted by professionals in the field of cyber security. The literature review starts off by introducing the motivations behind network attacks and gives reasons why a cybercriminal may want to target a network. They were: financial gain, industrial or political espionage, military warfare, activism or hacktivism. After this he Symantec report cyber security report 2015 was introduced as one of the aim areas of the literature review. The report examined all of the threats from 2015 and highlighted three key threat, they were: A new zero-day vulnerability was discovered each week, over half a billion personal records were stolen or lost in 2015 and major security vulnerabilities in three quarters of the most popular websites. The first point zero-day attack discussed the vulnerabilities of software and network bugs which could lead to exploitation. The report displays a number of graphs looking at the percentage of new vulnerabilities, zero day vulnerabilities and percentage of which were critical. The graphs show alarming results as there seemed to be a trend of gradual and some cases extreme increase in the number of zero day attacks. Following this academic research looked into recognizing and categorizing zero day attacks. The next second of the Symantec security report was discussing Stolen or Lost data. The report stated that over 191 million records were exposed and a total of 9 mega breaches occurred in the year 2015. Like with zero day attacks the report gives graphical data

to analyses and shows the number of total breaches were down by 2% but the breaches with more than 10 million identities exposed and average number exposed per breach had increased. Following on from this there was small section on to protect data in the cloud. The research suggested a number of ways to protect security physical through restricted access or with Intrusion detection systems with firewalls. The last part of the Symantec threat report discussed major security issues with websites. The paper highlighted that over 75% of websites are unpatched vulnerabilities while 15% percent were deemed critical while the technology industry was the largest target in the year 2015 for web based attacks. Following this the one of the main attacks was discussed SQL injection how it affected web based databases. Research into showed that SQL was very much alive and web based firewalls alone wasn't enough to protect data. The research suggested a combination of firewalls, intrusion detection and transparent view of security between developers of the website and security officials to keep the website patched and secure. Another web base threat was discussed in form of Denial of Service attack and the impact on user experience, service availability and network infrastructure. After this examples of SQL injection and DDoS attacks were given and the damage they produce if effective. The examples were the attacks on TalkTalk and the Carphone warehouse.

In part 2 of the literature review it looked into intrusion detect systems impartially Snort an open sourced version. The research centred around on the configuration, tweaking of Snort rules, CPU and RAM usage and placement in a network. The research deemed that a Windows platform was more suitable for this project as it handled moderate amounts of traffic better than Linux. Adjusting CPU and RAM values had an impact depending on what rules and signature set was being used. The second research paper discussed improving the ruleset for Snort for a better detection rate which is important for academics and the security industry. The final part of the research discussed the placement of Snort sensors in a centralized network how it could take advantage of the hierarchical model.

The third part of the research considered the methodology and reasoning behind different methods of research and the disadvantages of them so the reader was able to replicate the study. In this section the experimental hypothesis was discussed, the hardware and software and the different inputs. Then on why the experiment was reliable which was due to PCAPs which contained malicious software.

The fourth part of the research discussed the design choices made to create the test bed. The requirement specification a list of specifications that the design had to meet to achieve the set

aims. This included the experiment had to be A topology must be created to resemble key and mission critical services either physical or virtualized., some form of firewall must be setup to block any unnecessary traffic to prevent outside alerts effecting the experiments, Intrusion Detection/Prevention system must be implemented at key areas of the topology modes alert or drop packets from the experiments carried out, rules from Intrusion Detection/Prevention systems must be configurable to allow custom rule set from specific attacks and finally alerts generated must be logged and sort inform of a text document to examination later. After this the original design of the experiment was discussed which included Intrusion Detection system and Intrusion Prevention System which were Cisco based. Following this a section on why physical experiment was not feasible due to cost, licensing, lack of support via signature updates. Following this the virtual setup was discussed with Cisco IP/Ds devices how it could be implemented within GNS3. Problems occurred with crashing of the client and the losing of connection between the host due to the lack of support of switches. This put a nail in the coffin for trying to do a comparison between both Intrusion Prevention and Detection using Cisco and Snort devices. The final section of design looked at Snort topology which could be virtualized using a combination of open virtual box and open source technologies like pfsense a firewall and Windows server 2012 and how they could be tested via PCAPs which were guaranteed to produce results.

The next part of the project is the implementation phase where the experiment was setup and tested. In this section it shown screen shots of the installation of Snort and its rules. After this attacks like DDoS, Nmap and SQL injection were used and log was analyzed. The log showed that the rules were effective in detecting the attacks and alerting the network administrator. Each result of the attack was discussed and examined. Then it was discussed if the results were expected and how they linked back to the literature review. The final section of the report evaluated the conclusions which were found from the experiment and how they met the aim and objectives for the project and if the project was going to be repeated how certain procedures will be done differently.

The cyber security field has a vast number of research streams in which a project like this could sit and categorize itself. Using real time analysis by detecting network attacks is interesting and provides further area of research. Another area in which this project could pair is with disaster recovery of data and the prevention of further damage to other devices by tracing the attacker's footsteps which could be potentially interesting. On the other hand, this

project had loose ends in which intrusion prevention systems were not analyzed but were discussed. This could be a following avenue of research to build upon this project and it could add a lot more detail to this field.

Reflection

In the final section of the project it is time to reflect back on the project as a whole and discuss what was learnt and how it impacted my learning.

At the start of the project I was perhaps too ambitious to test both IPS and IDS technologies and the same time I believe this had a negative impact on my project by having to change my experiment and literature review half way through. But, I believe that I did the upmost to deliver the project as stated in aims and objectives. I learn how to approach the project from different angles by trying to test Cisco IPS under all situations as possible by trying different platforms like Virtualbox and GNS3 but I did not work due to crashing and errors. This impacted by belief in when ether the project could be achieved in some sort of fashion but looking from a different angle helped me reengineer the project to focus on Snort Intrusion detection. From research carried out in the literature review I learnt the pros and cons of using Snort in different situations, how to tweak and create my own rules from scratch. I am proud of myself for doing this. From this perhaps more background research before choosing a project could have shown me the difficulties in emulating Intrusion Prevention appliance either physically or virtually.

On this project organization was major pitfall and stopped me completing the project a lot earlier than expected due to failures in experimentation and completing the work to deadlines set by my project supervisor. This could have been avoided very easily by doing the work on time. I was happy in the way the project was tested and gave me results from NMAP, SQL and DDoS attacks which linked back to Symantec report. The integration of the Symantec Security Report 2015 gave my literature review structure and I could link related research back to report highlighting the importance of the research in context. Maybe for future projects it could be beneficial to compare different reports and look at their stats and figures to show any differences or links between them. I am proud that I looked at a lot of research and picked the necessary and relevant to back up the Symantec report and Snort Intrusion Detection.

Having not implemented and tested a IPS solution leaves my project with loose ends and the ability not to complete the set objects and aims. But, this can be a learn curve carefully choosing a project in a future to help to fix this error and opens the door for future research. I end this section having feeling let down, disappointed with myself for not completing the full projected stated in aims in objectives. The whole point of a major project is to push and

discover new research and things about one self and from this I am left disappointed. On the other hand, I am proud of myself for completing a major project on my own. To conclude, I have mixed feelings about this project from being proud and happy from completing the project left disappointed that the full aims and objectives were not achieved.

References

- Kaur, R.; Singh, M., "Efficient hybrid technique for detecting zero-day polymorphic worms," Advance Computing Conference (IACC), 2014 IEEE International , pp.95,100, 21-22 Feb. 2014.
<https://www.sans.org/reading-room/whitepapers/bestprac/defenses-zero-day-exploits-various-sized-organizations-35562>
- <https://www.fireeye.com/blog/threat-research/2015/06/operation-clandestine-wolf-adobe-flash-zero-day.html>
- <http://researchcenter.paloaltonetworks.com/2015/07/apt-group-ups-targets-us-government-with-hacking-team-flash-exploit/>
- https://users.ece.cmu.edu/~tdumitra/public_documents/bilge12_zero_day.pdf
https://www.symantec.com/content/dam/symantec/docs/reports/istr-21-2016-en.pdf?aid=elq_&om_sem_kw=elq_14872812&om_ext_cid=biz_email_elq_&elqTrackId=283a3acdb3ff42f4a70ab5a9f236eb71&elqaid=2902&elqat=2
- <http://ieeexplore.ieee.org.ezproxy.uwtsd.ac.uk/stamp/stamp.jsp?tp=&arnumber=7223168>
- http://delivery.acm.org.ezproxy.uwtsd.ac.uk/10.1145/2540000/2537884/p147-afek.pdf?ip=193.62.89.124&id=2537884&acc=ACTIVE%20SERVICE&key=BF07A2EE685417C5%2ED3B0B0D06D8A4908%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&C_FID=599740826&CFTOKEN=29441120&__acm__=1460504812_20b251075a7b54e0609ad122b1086d5c
- http://ac.els-cdn.com.ezproxy.uwtsd.ac.uk/S0167404814000182/1-s2.0-S0167404814000182-main.pdf?_tid=d14bccb6-0108-11e6-af45-00000aacb361&acdnt=1460504986_a0a399af0512f9ed2022729428ed0765
- <http://ieeexplore.ieee.org.ezproxy.uwtsd.ac.uk/stamp/stamp.jsp?tp=&arnumber=6542524>
- <http://ieeexplore.ieee.org.ezproxy.uwtsd.ac.uk/stamp/stamp.jsp?tp=&arnumber=6569317>
- <http://ieeexplore.ieee.org.ezproxy.uwtsd.ac.uk/stamp/stamp.jsp?tp=&arnumber=6974088>
- <http://ieeexplore.ieee.org.ezproxy.uwtsd.ac.uk/stamp/stamp.jsp?tp=&arnumber=6974088>
- <http://ieeexplore.ieee.org.ezproxy.uwtsd.ac.uk/stamp/stamp.jsp?tp=&arnumber=7237241>
https://www.owasp.org/index.php/SQL_Injection
- http://ac.els-cdn.com.ezproxy.uwtsd.ac.uk/S1877050915034985/1-s2.0-S1877050915034985-main.pdf?_tid=30199c4c-0680-11e6-a679-00000aacb35d&acdnt=1461106011_c4c6b8bc21133a2f3c2d374b92bfb1de
- <http://seclab.cs.ucdavis.edu/papers/tse96.pdf>

http://ac.els-cdn.com.ezproxy.uwtsd.ac.uk/S1877050915034985/1-s2.0-S1877050915034985-main.pdf?_tid=30199c4c-0680-11e6-a679-00000aacb35d&acdnt=1461106011_c4c6b8bc21133a2f3c2d374b92bfb1de

http://delivery.acm.org.ezproxy.uwtsd.ac.uk/10.1145/2670000/2668327/a5-papalambrou.pdf?ip=193.62.89.124&id=2668327&acc=ACTIVE%20SERVICE&key=BF07A2EE685417C5%2ED3B0B0D06D8A4908%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&CFID=602982333&CFTOKEN=87174046&__acm__=1461106549_b9b4a378f83e6bf2c545464d3fdf5cd2#URLTOKEN#

http://ac.els-cdn.com.ezproxy.uwtsd.ac.uk/S1084804509001040/1-s2.0-S1084804509001040-main.pdf?_tid=c8af7964-06f7-11e6-886a-00000aacb35f&acdnt=1461157377_cd42d51a796962dacc75d60dec79cfb0

<http://ieeexplore.ieee.org.ezproxy.uwtsd.ac.uk/stamp/stamp.jsp?tp=&arnumber=6914042>

<http://ieeexplore.ieee.org.ezproxy.uwtsd.ac.uk/stamp/stamp.jsp?tp=&arnumber=6188084>

<http://ieeexplore.ieee.org.ezproxy.uwtsd.ac.uk/stamp/stamp.jsp?tp=&arnumber=6188084>

