

A RESEARCH OF INTRUSION DETECTION SYSTEM IN
GENERAL AS WELL AS SNORT IN DETAIL BEFORE
INTRODUCING IT INTO GLOBAL CYBER SOFT
COMPANY

Mohamed Aly Pasha

A thesis submitted in partial fulfillment
of the requirements of the
Vietnamese-German University
for the degree of
Master of Science

Research undertaken in the
Business Information Systems Department

November 2016
(This page intentionally left blank)

(This page intentionally left blank)

Disclaimer

I certify that the attached thesis is my original work. No other person's work has been used without due acknowledgement. Except where I have clearly stated that I have used some of this material, it has not been presented by me for examination in any other course or subject at this or any other institution.

I understand that the work submitted may be reproduced and/or communicated for the purpose of detecting plagiarism. I understand that should this declaration be false; I am liable to be penalized under the Vietnamese-German University regulations.

Student's signature

Date:

Abstract

A research of intrusion detection system in general as well as Snort in detail before introducing it into Global Cyber Soft Company

Mohamed Aly Pasha

Nowadays, with the expanding of Internet all over the world, many hackers who try to steal some confident information to use for their own benefit or just destroy or modify it. From this, IDS/IPS (intrusion detection/prevention system) was born to help users, companies or institutions to detect and prevent this issue. My company currently is considering to install an intrusion detection system and this thesis is aim to study IDS in general to get the common knowledge and Snort in detail for our managers make a decision to implement it or not. In this Master's thesis, I will cover two main parts: the first one includes of an intense research study about the IDS/IPS and comparing the differences between them. Subsequently, we will continue this part with some limitation points where IDS/IPS still needs to be checked and show up some requirements for the better intrusion detection system.

Second part of the thesis would try to show how to configure of Snort in Windows environment. Some demo of attacking on the environment where Snort is installed will be performed in order to show its protection on reality. Finally, we will evaluate some advantages and disadvantages compared to other software to conclude whether we should implement it on our company or not.

Keywords: - intrusion detection system, prevention system, Snort, IDS, NIDS, Firewall, Suricata, False positive.

Acknowledgements

First and foremost, I would like to thank my supervisor, Prof. Dr. Friedbert Kaspar, without his instruction and dedicated support through each step of the process, this thesis would not ever be accomplished.

I also would like to express my special gratitude to my second supervisor Prof. Dr. Tomas Benz and all professors from Heilbronn University and Furtwangen University, who have been flying back and forth between Germany and Vietnam to lecture me over the course' years. I really appreciate their effort and enthusiasm in every given lecture which passionate and inspire me during the master course of business information systems.

Finally, none of this could happen to me without the strong backend support from my family, I would like to thank all of my beloveds for their advices and being always with me.

Table of tables

Disclaimer.....	1
Abstract.....	2
Acknowledgements.....	3
Table of tables.....	4
Table of Figures	6
1. Introduction.....	7
1.1 Computer security background.....	7
1.2 Some security terms.....	7
1.2.1 Intrusion detection system (IDS)	7
1.2.2 Firewall	8
1.2.3 Intrusion prevention system (IPS).....	8
1.2.4 Snort.....	8
1.2.5 False Positive	9
1.2.6 True positive	9
1.2.7 False Negative.....	9
1.2.8 True Negative.....	9
1.2.9 Alert	9
1.2.10 Logs	9
1.2.11 Sensor	9
1.2.12 Signature.....	10
1.3 Purpose & research questions	10
1.4 Organization of the thesis	10
2. Literature review and thesis contribution	11
3. IDS overview	12
3.1 History	12
3.2 IDS Architecture.....	13
3.3 IDS classification.....	14
3.4 Where to place an IDS	20
4. Additional tools support for IDS	23
4.1 Port scanners.....	23
4.2 Application scanners.....	24
4.3 Vulnerability scanner.....	24
4.4 Honeypot.....	25
4.5 Padded cell.....	29
5. Some concerns of information security and the future of IDS	30

5.1 CIA triad of information security	30
5.2 Comparing IDS with firewall	31
5.3 Types of Computer Attacks Commonly Detected by IDSs.....	31
6. Snort.....	36
6.1 What is Snort	36
6.2 Snort Architecture.....	36
6.3 Installation and configuration	38
6.4 Compare Snort and Suricata	50
6.5 Improve your custom Snort's rules.....	52
7. Conclusion	54
8. References.....	56
Appendix A: Acronym	65

Table of Figures

FIGURE 1 - A SAMPLE IDS	13
FIGURE 2 - IDS COMPONENTS	14
FIGURE 3 - HIDS	15
FIGURE 4 - NIDS	16
FIGURE 5 - IDS IN FRONT OF EXTERNAL FIREWALL.....	21
FIGURE 6 - IDS IN THE DMZ	21
FIGURE 7 - AN EXAMPLE OF AN INSTALLED HONEY POT SYSTEMS	26
FIGURE 8 - TYPES OF HONEY POT	26
FIGURE 9 - 1ST GENERATION HONEYNET	28
FIGURE 10 - 2ST GENERATION HONEYNET	29
FIGURE 11 - DIAGRAM SHOWS A SCANNING ATTACK WHERE A SINGLE ATTACK HOST SCANS A NUMBER OF VICTIMS	32
FIGURE 12 - DIAGRAM SHOWS A DENIAL OF SERVICE ATTACK (DDOS IN THIS CASE)	33
FIGURE 13 - COMPONENTS OF SNORT.....	36
FIGURE 14 - INSTALLATION OF WINPCAP	38
FIGURE 15 - INSTALLATION OF SNORT	40
FIGURE 16 - ZENMAP TOOL	48

1. Introduction

1.1 Computer security background

According to intelligence reports on the global threat by the US company CrowdStrike, Việt Nam ranked number 1 in the world for the rate of malware infection through portable storage devices (USB, memory sticks or external drives), with 70.83 per cent of PCs infected and 39.95 per cent of users faced with malware from cyberspace. In the first nine months of 2015, 18,085 Vietnamese websites, including 88 state-run platforms, were infected with malicious software, according to the Việt Nam Computer Emergency Response Team (VNCERT). [1]

System admins nowadays have to deal with not only monitoring pro-actively all the service systems but also react promptly to different attacking from outsiders. We cannot measure all consequences when attackers can control a company's information system. Especially, if there are main backbone systems such as e-government, banking, satellite system or big e-commercial chain system which under hacked by black hat hacker can cause remarkably damage on reputation and economic side. An IDS/IPS (intrusion detection/prevention system) is the software / hardware device that can collect information from the whole system to inform or alert of the presence of current attack or possible intrusion.

Nowadays, IDS/IPS (intrusion detection/prevention system) more and more becomes popular and act as main role in any security and safety information policy on almost firms. IDS now is really diversity and abundant, including software or hardware solution. Overall IDS main functions are detection, defenses, and most importantly is quickly hinder malicious attacking. In these prominent software, Snort has raised to be a must-choice product for system admins as well as small and medium enterprises.

1.2 Some security terms

In order to make the next chapters more understandable, the most common words and definitions of the network field will be explained

1.2.1 Intrusion detection system (IDS)

Intrusion detection system (IDS) is software or hardware product to inspect and alarm any inbound or outbound network activity and unauthorized access or malicious

activities to protected host or network system. It will be explained more detail in chapter 3.

1.2.2 Firewall

Firewalls are software programs or hardware devices that filter the traffic that flows into you PC or your network through an internet connection. They sift through the data flow & block that which they deem (based on how & for what you have tuned the firewall) harmful to your network or computer system.

When connected to the internet, even a standalone PC or a network of interconnected computers make easy targets for malicious software & unscrupulous hackers. A firewall can offer the security that makes you less vulnerable and also protect your data from being compromised or your computers being taken hostage. [2]

1.2.3 Intrusion prevention system (IPS)

According to Karen Scarfone [3], network intrusion prevention systems (IPS) monitor and analyze an organization's network traffic to identify malicious activity and optionally -- stop that activity by dropping and/or blocking associated network connections. IPS have been used for many years at key network locations, such as in close proximity to firewalls to identify a variety of network-based attacks that other security technologies are unable to detect.

The predecessor to network intrusion prevention systems, known as intrusion detection systems (IDS), provide the same types of functionality, except IDS cannot stop malicious activity.

Firewall, IDS, IPS function independently and support each other to protect the internal network. For example, firewall is like your lock at your house, IDS is the camera and IPS is your dog. Firewall will block all illegal access; IDS warns us about incoming attack and IPS would react on incoming attack by trying to fight it back.

1.2.4 Snort

It is an open source intrusion prevention system capable of real-time traffic analysis and packet logging.

1.2.5 False Positive

In IDS, a false positive is count when detection engine generates an alert for an event that is not malicious.

1.2.6 True positive

A potential attack which triggers to produce an alarm

1.2.7 False Negative

A failure of an IDS to detect an actual attack, so allowing it to enter the network without any notice.

1.2.8 True Negative

When no attack has taken place and no alarm is raised.

1.2.9 Alert

Alerts are any sort of user notification of an intruder activity. When an IDS detects an intruder, it has to inform security administrator about this using alerts. Alerts may be in the form of pop-up windows, logging to a console, sending e-mail and so on. Alerts are also stored in log files or databases where they can be viewed later on by security experts. [4]

1.2.10 Logs

The log messages are usually saved in file. Log messages can be saved by text or binary format. [4]

1.2.11 Sensor

Intrusion Prevention Systems are based on sensors, which recognize abnormal activity. There sensors can be arranged in two classes as network and host based. Network sensors can be assigned in channel communication gap for analysis of passing packets. Analysis of packets are based on signature or behavioral methods. Host based sensors can be used to detect both remote and local attacks.

1.2.12 Signature

A signature uses to detect a kind of attack by looking at specific patterns. For example, when a packet sent to a web server contained “scripts/iisadmin” pattern then it wants to signal “may the web server can be attacked”. However, it’s impossible to detect new attacks due to no known pattern.

1.3 Purpose & research questions

The main ideas of this thesis are to provide information of IDS/IPS and Snort to implement it in real environment in GCS company. In this research, I try to find out what Snort is and the benefit when apply it in project. After that, I will show the config results after research as well as some advantages and disadvantages of which we faced when implementing Snort in simulation environment.

To get deep knowledge in IDS and Snort, I formulate some research questions regarding them as follows:

- Why does the company network need IDS?
- What are limitations and strong points of a IDS system?
- What is the factors or problems impacts on a company’s network when applying Snort?
- What are the pros and cons of Snort compared to other tools?
- Can we improve Snort’s rule to enhance its protection?

1.4 Organization of the thesis

The thesis begins with computer security background then clarifying some network terms before followed up by literature review in chapter 2.

Some things related to the overview of IDS: history, why we need IDS, classification, architecture and model are briefly described in chapter 3.

In chapter 4, about introduction of some security tools can help to supplement the task of intrusion detectors.

Chapter 5 is the analyze of limitations and strong points of IDS compared to other type of defense such as firewall, antivirus.... In chapter 6, the thesis depicts the Snort detail: model, history, how to implement it on a windows computer. Also, adding some database, server tools connect to Snort to make it more power. Finally, the thesis ends up with conclusion, suggestions and future works in chapter 7.

2. Literature review and thesis contribution

Recently, almost firms or companies have been operating based on a network within their systems. Many types of risks can attack the computer systems and networks. This gives promote to IDS. IDS include protecting the network system information effectively by preventing, detecting and responding to attacks. However, there is no one hundred percent guarantee that these threats will not happen (Xu & Ning, 2008) [5].

IDSs are commonly observes packets to recognize their intrusive behaviors. But according to Xu and Ning [5], IDS is unable to prevent attacks totally; however, it collects information when threats are met. Maggi et al [6] suggests that this collected information can be used to mend gaps and mistakes in the system network. When Intrusion Detection System discovers a suspicious activity within the network, it sends an alert to the network administration about the potential threat that might be an intrusion attempt.

In my company system network, currently no IDS system is set up and vulnerable to attacking. By the fact that, we will need to research about IDS before deploying it to our network system. Yet, we conclusive to use Snort as an example for an IDS in this thesis. In addition, this thesis aims for answering the main research question that we mention above in 1.3 section. But Snort could be deployed on my company or not depends on how it meets the requirements of my company's need as follow:

- ✓ It should be constantly updated, which is a key part of the deploy decision
- ✓ It should be designed to merge smoothly into the current network system of the enterprise
- ✓ It features to detect critical threats, perform behavioral analysis
- ✓ It should be integrated with the firewall
- ✓ Capable of signature matching and behavior anomaly detection
- ✓ Capable of detection on high speed analysis

There exist many different scientific articles describing IDS, and an effort will be made to select the most important, relevant and newest ones. Although the scientific articles are the key distribution for this thesis, there also exist some relevant web pages and slides. The places where the relevant articles will be found are Google search engine and other papers, thesis... shown in references part.

3. IDS overview

3.1 History

James P. Anderson [7] is the first person who mention to IDSs on his paper “Computer Security Threat: Monitoring and Surveillance” in 1980. He discerned security threats in two kinds of external and internal ones based on the access permission which user can access to the system or not. He focused on the collection of records that showed abnormal use of the system, abnormal frequency of use, abnormal patterns of data. He also alerted about the problem of the legit user that maybe has access to confidential internal data; it would be very difficult to record a feasible trail to detect some misuse.

His work had been continuing in 1987 by Dr. Dorothy Denning who engaged on a project to detect some possible misuse by taking records of the computer activities of all the users.

After that, Intrusion Detection Expert System (IDES) was developed from 1984 to 1986 by Denning and Peter Neumann, funded by the Navy of U.S [8]. Everything was based on the study of user profiles that would show some evidences in case of abuse or misuse.

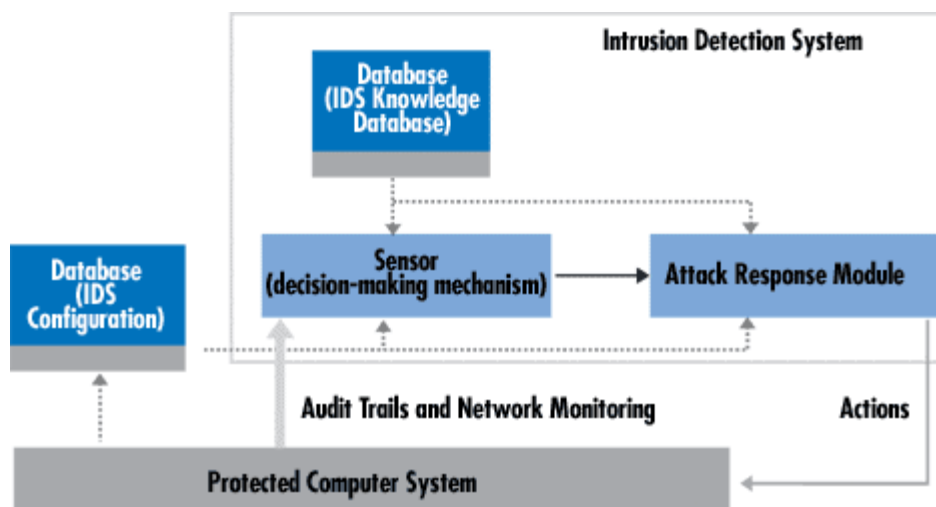
However, the first IDS that monitored network traffic was the Network System Monitor (NSM) and was developed in the California University to work on an UNIX computer of Sun. The mode of use was very close to the IDS of today. In the early 1990s, intrusion detection development for commercial emerged and Stalker was the first commercial product was developed by Haystack Labs followed by Netranger of Wheel Group in 1994. [9]

During the last decade, various vendors such as Cisco, Internet Security Systems, Symantec... have been continuously developed their IDSs for both enterprises or even home-users due to the exponential growth of the internet and the need of secured network system. Snort, an open source Network IDS, was launched by Martin Roesch

in 1998 [10], that will be studied in depth in the following chapters. Next year, Okena Systems introduced the first Intrusion Prevention System (IPS) with name “Storm Watch”. IPS are the systems that detect the intrusions and are able to react on alarming situation and co-operate with firewall without any intermediary applications. Therefore, a wide range of offerings can be found, from costly products to very good open source ones nowadays. [11]

3.2 IDS Architecture

According to Przemyslaw Kazienko & Piotr Dorosz [12], an intrusion detection systems always a center element: a sensor (an analysis engine) which comprises decision-making procedure. The sensor is responsible for detecting intrusions. Sensors collect crude input data from three main sources such as depicted in Fig.1: IDS knowledge database, IDS configuration and audit trails. This information creates the basis for a further decision-making process.



(Fig.1) A sample IDS. The arrow width is proportional to the amount of information flowing between system components [13]

The sensor is integrated with an event generator which responsible for data collection (see Fig.2). The policy of event generator decides how the collection method works by specifying the filtering mode of event notification information. The event generator (operating system, network, application) makes an information collection policy set of events that may be a log (or audit) of system events, or network packets. This collection can be stored either in the inside or outside data storage. In our

certain cases, no data storage is existed so the event data streams are transferred directly to the analyzer. [12]

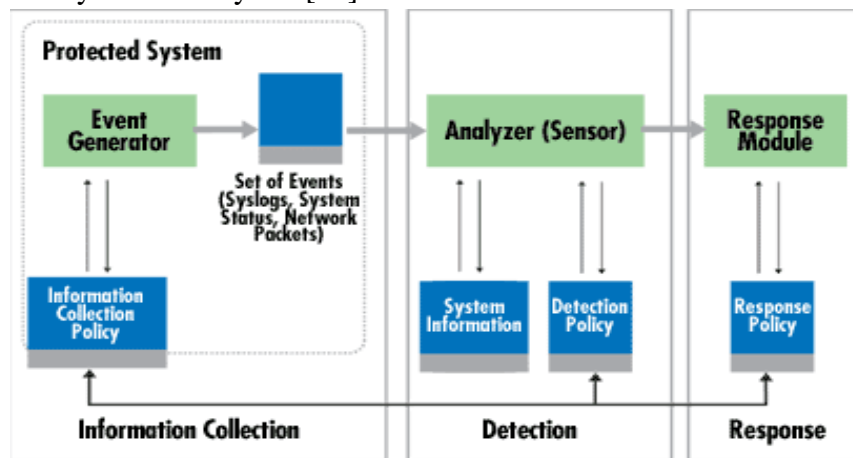


Fig.2 IDS components [14]

The role of the sensor detecting suspicious activities by filtering information and reject any unrelated data obtained from the event set associated with the protected system. The analyzer uses the detection policy database for this intention. The detection policy database is made up of the following elements: attack signature patterns, normal behavior profiles, IDS configuration parameters.

Intrusion detection systems can be arranged as either centralized (for example, built-in IDS within a firewall) or distributed. A distributed ID is composed of many Intrusion Detection Systems (IDS) placed span on a large-scale network in internet that exchange information with each other. More complex systems follow an agent structure principle where small autonomous modules are organized on a per-host basis across the protected network [15]. The task of the agent is to observe and separate out all activities within the protected area and make first analysis and even take on a response action. One of the most important components of intrusion detection systems is the cooperative agent network. Moreover, agents can be specifically given to task detecting certain known attack signatures. This is a decisive factor when introducing protection means associated with new types of attacks [12] [16] [17] [18].

3.3 IDS classification

Mr Abhishek Pharate, Harsha Bhat and Vaibhav Shilimkar in their “Classification of Intrusion Detection Systems” paper [19] supposes there are four ways to classify

IDSs depending on location, functionality, deployment approach, and detection mechanism. These ones are the most common criteria and they will be explained in more detail in the following pages:

3.3.1 Location:

3.3.1.1 Host Based Network Intrusion Detection

HIDS is single anti threat applications (firewalls, antivirus software and spyware-detection programs) which monitors and analyze the security of internal system as well as protect it from external attacks. The detection of internal attacks using OS auditing mechanism which detects which program access which resource and is there any security breach. For example, word-processor suddenly starts accessing system password database and starts modifying it. In external attacks, a HIDS analyzes the traffic to and from the specific computer on which the intrusion detection software is installed. HIDS analyses packets to and from that system (computer) on its network interfaces. HIDS answers by logging the activity and informing about it to designated authority. An example of HIDS is PortSentry.

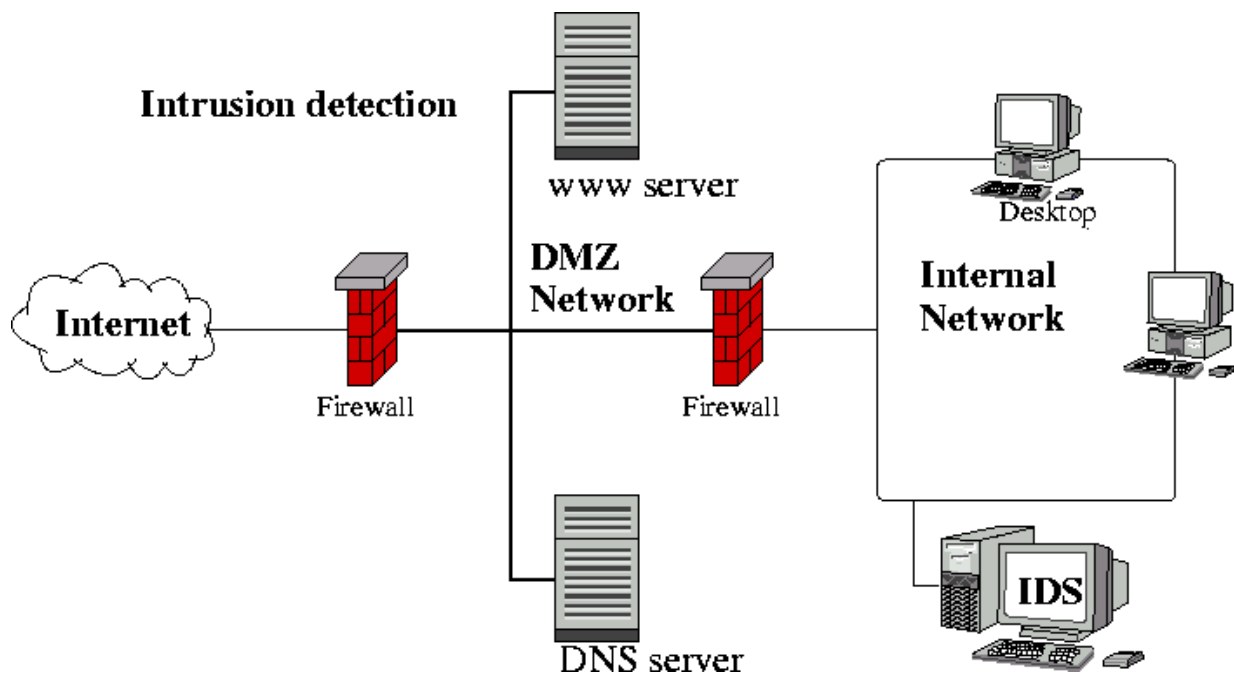


Fig3 HIDS [20]

Advantages:

1. Cost effective for a small LAN having few hosts.
2. Can analyze encrypted traffic.
3. HIDS does not requires much extra hardware as install on a single host.
4. Can verify if an attack successful or not, while NIDS only give alert.

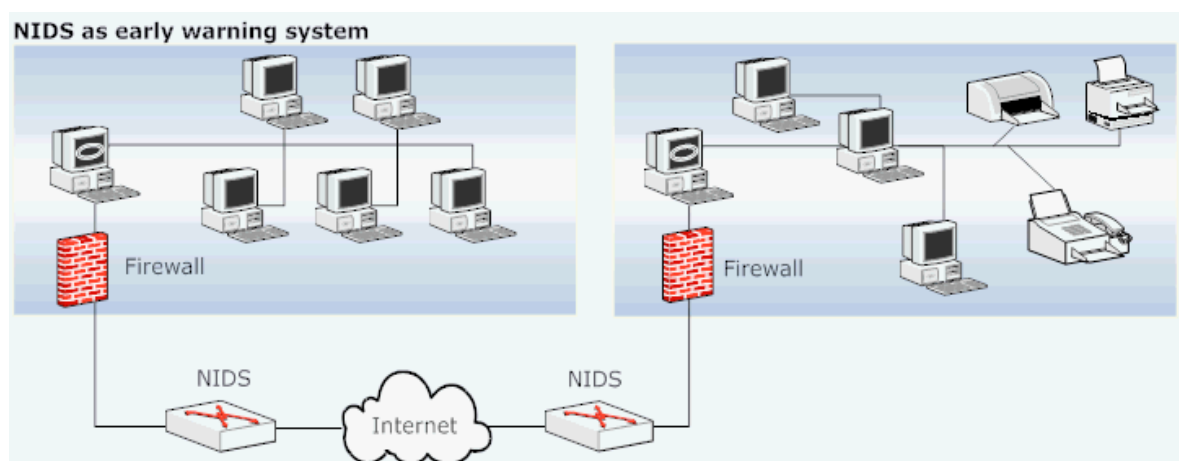
Disadvantage:

1. Not protect entire network, ineffective for large -scale attack
2. Consume power, resource on installed host server
3. Ineffective on DDOS attack
4. Can be disabled as soon as installed host server is attacked

3.3.1.2 Network intrusion detection system

Network intrusion detection system (NIDS) monitors network traffic and analyzes the passing traffic for attacks on a planned point in a network, often the entry and exit point of a data from inside network to outside. On the event of an attack or a peculiar behavior is detected, an alert can be sent to the system administrator. NIDS can detect 4 major types of attacks: denial of services, probe, user to root and remote to user. Examples of NIDS implementation are Snort, Cisco Secure IDS and Dragon Enterasys. [21]

Fig4 NIDS [22]



Pros:

1. OS independent.
2. Against DDOS attack.
3. Protect large scale network

Cons:

1. Cannot scan encrypted content.
2. Ineffective on high speed network.
3. Failure rate is higher due to it based on pre-defined attack signatures.

3.3.2 Functionality:

3.3.2.1 Intrusion Detection System

Intrusion detection system is a software/hardware which monitor if there is any malicious activity or policy violation targeted to computing and network resources then reports it to administrator. There are two types intrusion detection techniques:

- Statistical anomaly detection
- Misuse detection.

Anomaly detection analyses information gathered and created a baseline behaviors. It tries to find out users' or system's abnormal behavior by compare it to the baseline behavior. When the deviation from normal behavior is detected, the IDS trigger its alarm for intrusion. Misuse detection is based on a database of signature for known attacks and exploits. It would trigger alarm during the analysis of network packets if it finds any pattern match to one of those known attack patterns. Since there are many networks based exploits coming on each month, the need to be updated database frequently. IDS use different algorithms such as Adaptive Resonance theory, Self-organizing m p, and genetic algorithm. [23] [24]

3.3.2.2 Intrusion Prevention System

IDS was only capable of detection of intrusion without prevention action. Intrusion Detection System Proactive technique which prevents attack before entering a network by examining packets and their pattern and dropping the suspected packets. IPS is active and smart and system which provides early detection attack. IPS works on 2, 3,

4 and 7 layer of OSI. IPS has the functionality of doing activity of Early Detection, proactive technique, early prevent the attack, when an attack is identified then blocks the offending data. [25]

3.3.2.3 Intrusion Detection and Prevention system

Firewall itself does not help against network attacks such as Denial of service attack on open ports. So IDPS could be used to protect network services with firewall. Mainly there are three parts in IDPS:

1. preprocessing
2. classification
3. protection.

In preprocessing part, packet sniffer is used capture packet information from the Ethernet interface card. Then each of the preprocessed data is classified as mainly two types: attack data and normal data. This information is passed to protection part which takes suitable action according to type of packet for prevention, in case of DDOS, it'll block or drop all packets from the attacker IP. [26]

3.3.3 Deployment approach:

3.3.3.1 Single host

In single host deployment of network intrusion detection system, the system is installed on a single host in network that may be a router, a server or network switch. The whole traffic goes through the network via that node where it is monitored for attacks and normal data by the NIDS. Examples of single host intrusion detection systems is NetRanger, Tripwire.... [19]

Pros:

1. A single NIDS can observe a large network
2. Its boundary is limited and it just listens

Cons:

1. It is hard to process all packets in a high-speed network.

3.3.3.2 Multiple host (Distributed agents)

In distributed deployment of NIDS, the system is installed on some nodes in the network as NIDS agents. These agents then watch the traffic that is traversed through that specific node and generate results to the NIDS management system. This NIDS management system coordinated with the agents and generates alarms for appropriate packets and broadcasts it on network. Examples of multiple host intrusion detection system is AAFID, NIDES, Stalker. [27] [28] [29] [19]

Pros:

1. Multi NIDS is capable of processing all packets.

Cons:

1. It is more complex to manage and must be set-up properly for each different host.
2. It's hard to incorporate between NIDS agents.

3.3.4 Detection Mechanism:

3.3.4.1 Signature based

In signature based detection mechanism the attack patterns are saved in the database. Each packet of the network traffic is measured with the attack patterns to detect peculiar behavior. Signature based intrusion detection system discovers only known attacks in database. For examples Suricata is a signature based intrusion detection system. [30] [19]

Pros:

1. It has low false positive with attack signatures are clearly defined.
2. Simple to use.

Cons:

1. The collected data could be out of date.
2. Hard to detect not-in-database attacks.
3. Alert whenever having an attack regardless of outcome.

3.3.4.2 Anomaly based intrusion detection system

Anomaly based intrusion detection system is based on the network behavior. The network behavior is specified by the system admin or is learned by through the training phase of the development of IDS. Rules are defined for both normal behavior and abnormal behavior. Snort and Bro-IDS are anomaly based intrusion detection system. [31] [19]

Pros:

1. It's able to detect unknown attacks.

Cons:

1. It's hard to define the rule set for intrusion detection.
2. It strongly depends on fitness of the rules.

3.4 Where to place an IDS

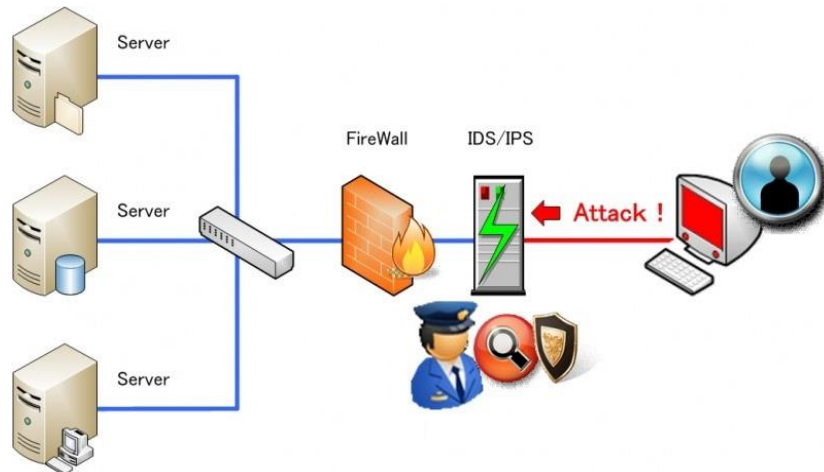
There are various ways to implement the IDS tools on our network system which has its own advantages and disadvantages. The best choice would be a reconciliation between cost and desired needs of the organization.

For this reason, the positions of the IDS within a network provide different characteristics. Noonan W.J. (2004) proposes eight recommendation locations for set IDS sensors [32], but I conclude from his idea there are four critical points that we need setup IDS/IPS to protect our system:

1. In front of your external firewall: Although having some arguments to the point of value of placing a sensor outside your firewall, system admins can gain some kind of reduction of traffic such as high volume denial of service events. The sensor should be configured to monitor the traffic on the switch

port that the firewall is connected to. When we implement the IDS/IPS in this location, we should significantly tune down the alerts because much of this traffic should be blocked by the firewall.

Fig5 IDS in front of external firewall [33]



2. Behind each firewall that provide access to DMZ: This is the most favorable location for sensor placement because these network connections represent main roads where all traffic between network modules must travel through. The sensor should be configured to monitor all the traffic on the network that the firewall is connected to. Some advantages when we put IDS on behind each firewall:
 - Observe attacks that penetrate the network's perimeter defenses from outside to server reside in this DMZ commonly.
 - Highlights problems with the network firewall policy.
 - Even if the incoming attack is unrecognized, the IDS can sometimes learn the outgoing traffic that results from the compromised server.

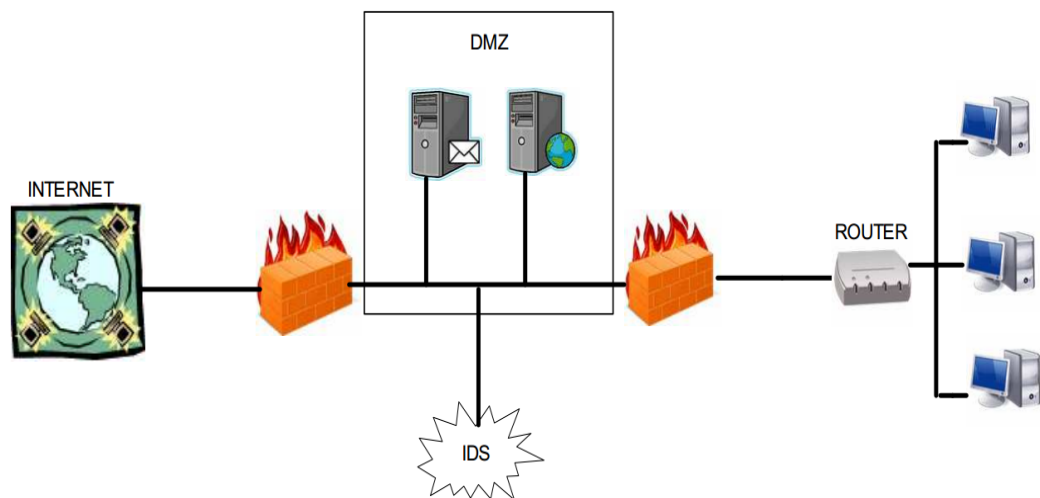


Fig6 IDS in the DMZ [34]

3. Placing a sensor at major network backbone point which usual traffic is moving in or out of the network segments that contain your most sensitive data:
 - a. Integrated into your gateway devices such as firewalls which have built-in IDS and IPS functionality. This is the location where most IPSs should be installed, because the firewall is already providing blocking functionality, and the IPS function merely enhances that functionality.
 - b. Behind your VPN: VPNs often provide an easy way for malicious traffic to enter your network. The reason is that often the remote connection's level of security is not the same with your internal network. Get in mind to place the IDS/IPS sensor behind the VPN so that it can monitor the unencrypted traffic as it passes to and from the VPN connections. You cannot collect any useful information if you place the IDS/IPS in front of the VPN concentrator because the data is encrypted.
 - c. In front of your server segments: Because your servers generally contain the most valuable data in your environment, you should implement an IDS/IPS in front of your server segment to detect

unauthorized activity by unauthorized users outside the organization's security perimeter. The easiest way to do this is to configure the monitoring interface to monitor the traffic on the switch port that connects the server segment to the rest of the network.

- d. On any critical subnets that contains or connects to critical resources: This is the generic place to catch all important traffic. We should implement sensors anywhere in our system network where we place valuable resources or want to know what kinds of traffic being passed.

4. Additional tools support for IDS

Intrusion detection systems cannot ensure completely security; therefore, the research on some security tools can help to harden the task of intrusion detectors. In this chapter, will described some mechanisms and security tools that having a closely relationship with the intrusion detection.

4.1 Port scanners

Most system is defenseless to port scanning so the best offense is a good defense. Most default installations have large ports open to allow outside connection. The bigger number of services that your system offers, the more undefended open for the system.

The scanned host can be tricked into replying by using a TCP (Transmission Control Protocol) port scanner such as Nmap to probe the number of open ports in a system. The easy way and easily detected is a basic TCP connect scan. The scanner tries to set up an official connection with a three-way handshake. The listen port will welcome the remote connection, and closed port will get a failure.

TCP half open connection means that is doesn't close the open connection. It'll sends a SYN segment to the targeted host and wait for reply. If the host answers with a RST (reset) then the port are closed, otherwise it will answer with a SYN/ACK (Synchronise/acknowledge) and it is opened.

TCP FIN is hardly undetected by most firewalls, packet filter, and scan detection programs. It will send a FIN packet to targeted host and wait for answer. The closed ports will answer with a RST and the open ports will ignore it. The attacking system simply take notes on ignored ports. So, the query is: How do we restrict the information our systems will show out. One solution to restrict the information got from port scans is to close non-essential services on the targeted systems, i.e. http should be the only service offered if you have a web server on your system. (Johan Nilsson,2006) [35] [36]

4.2 Application scanners

There are application scanners to be used for evaluating the security configuration of a specific application or services like Web, database and network domains that is hard to configure. Nikto is an example of a Web server scanner which seek for potentially dangerous files and CGI problems on servers. Database scanning can be done by Shadow or Metacoretex among others. [35]

4.3 Vulnerability scanner

A vulnerability scanner works such as a port scanner and tries to assesses security holes in networks or host systems and create a set of scan results by exploring all the hosts running in the defined IP range. The scanner tries to find all opened ports and corresponding services on all active hosts when the hosts have been found. In most scanners, there is a possibility of setting different scan modes and also set up which ports to scan. The scanner find the weak points in the scanned host by comparing running operating systems and software applications with known vulnerabilities stored in a database. [37]

There are two types of vulnerability scanners: host based and network based. A host based scanner is installed on every system that should be examined. It can be run as standalone software or be linked to a central part on the network. Host assessment software makes the vulnerability analysis from the inside of a host and looks for insecure file permissions, missing software patches, noncompliant security policies,

backdoors and Trojan horse installations. These features make it a preferred tool to use in security critical systems since the testing is time consuming and not scalable but provide good security supervision. [35]

The network based tools finds all hosts in the supplied IP range and analyses them for potential vulnerabilities. The network based tools are easier to use because they give information collected in one report summarizing the security situation ranging from one host to a large complex network. The disadvantages with the network vulnerability scanners is that they might not detect active hosts if there is a firewall running, not detecting open backdoors or not being able to perform certain tests due to disruption of normal services. [35] [38]

A network vulnerability assessment starts with determining which systems in the defined range that can be defined as online and accessible. Generally, scanners try many ways to trigger a response. If the host answers it will be put in a list of valid hosts. Most vulnerability scanners have an option for force scanning of a specific address regardless if a response is received or not. Different scanners use different ways of assessing if an address corresponds to an online host but the most common is the use of ICMP echo request (ping). Nessus as well as other scanners also has the ability to use both TCP and UDP packets to find out if the host is active. [35]

4.4 Honeypot

4.4.1 Why do Honeypots improve network security?

A honeypot is a computer connected to a network which used as a trap-set server (decoy) to examine security holes of the operating system or network by seeking to get unaccredited access into your network systems. Remember that Honey Pots do not take the place of other standard Internet security systems; they are unique tools to learn about the tactics of hackers.

Honey Pots can be setup at preferable place such as inside of a firewall or

inside/outside of the DMZ or even in all of the locations. In brief, Honeypots can be considered as a decoy as part of standard Intruder Detection Systems (IDS) but focus farther on deception-information gathering. [39]

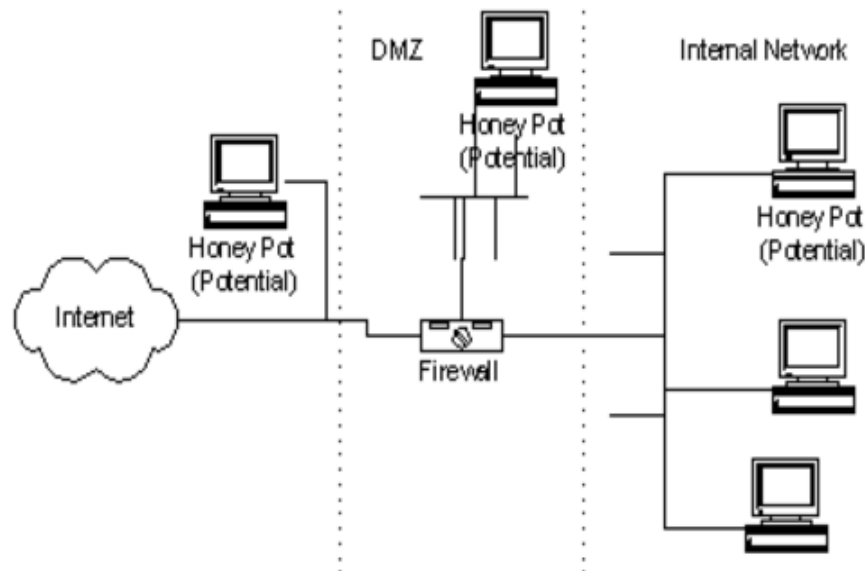


Fig7: an example of an installed Honey Pot systems [39]

4.4.2 Type of Honeypots

Honeypot consists of two main categories: low interactivity and interactive

+ Low interactivity: are simulated services, applications, and operating systems. Low level of risk, ease of deployment and maintenance, but the service is limited.

+ High interactivity: are the real services, applications and operating systems. Gather a large of information but risky and time-consuming for operating and maintaining.

Fig8: Types of Honey Pot [40]



- BackOfficer Friendly (BOF): A type of Honeypot is very easy on operation and configuration and can operate on any version of Windows and Unix, but only interact with certain services as simple as FTP, Telnet, SMTP ...

- Specter: It is kind of low interaction Honeypot but better interoperability than BOF, working on port 14, can alert and remote management. However, the specter BOF same restricted number of services and inflexible.

- Honeyd:

+ Honeyd listening on all TCP and UDP ports, these simulation services are designed with the aim of preventing and record attacks, interact with the attacker as a victim system.

+ Honeyd can simultaneously simulate many different operating systems. Currently, there are multiple versions of honeyd and can simulate approximately 473 operating systems but quite out-of-date now.

+ Honeyd is a low interaction honeypot which has many advantages however honeyd's drawback is unable to provide an actual operating system to interact with hackers and without having a warning mechanism to detect compromised or in-danger systems.

Honeynet is a high interaction honeypot. Honeynet provides real systems, applications, services. According to the “Honeynet Project”, there are two basic architectures of honeynets, Gen I and Gen II [41]. Gen I is simple architecture was the first developed. A network is placed behind an access control device, usually a firewall which separates the network into three different parts: Honeynet (in yellow), Internet, and system network. Also, any packet travel in or out of the Honeynet has to go through both the firewall and the router. The firewall is the primary tool for controlling the inbound and outbound connections whereas the router is designed to allow any inbound connections, but to limit those that are outbound. [42]

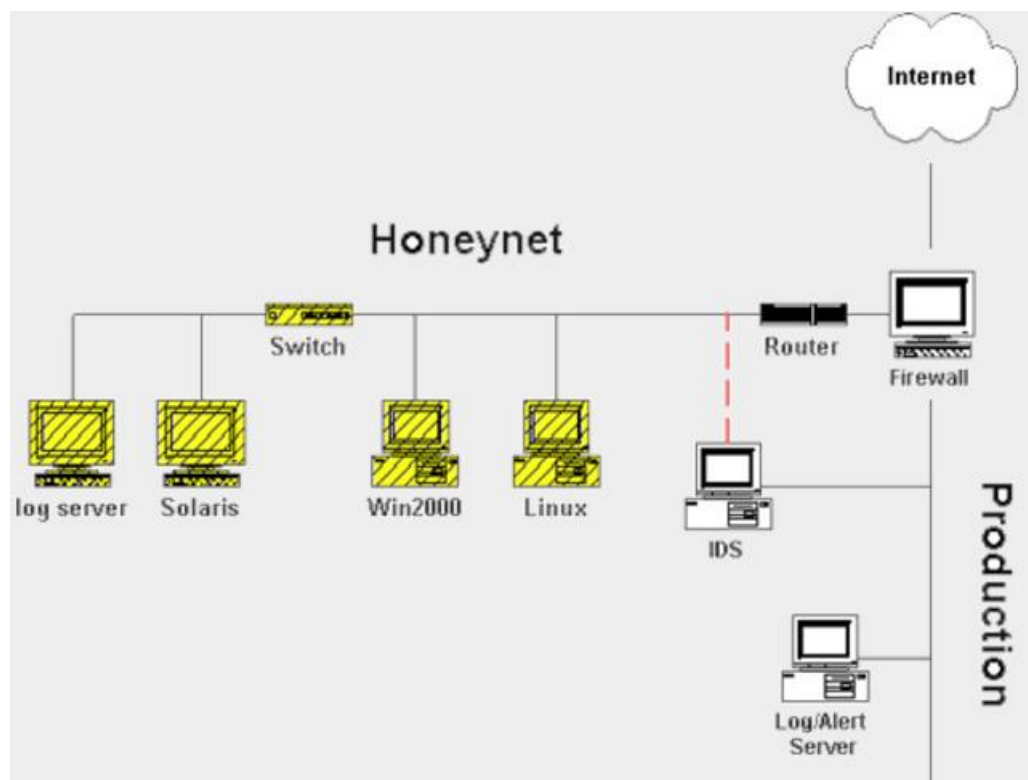


Fig9: 1st Generation Honeynet [41]

The aim of Gen II is to create a honeynet which is easier to deploy, while also being more difficult to detect. The deploy and managing will be much easier when all the requirements have been united into a single device. The most important element of a Honeynet is the gateway that is what separates a Honeynet from the Internet. This gateway is called a Honeywall simply because it acts like a wall. All traffic, inbound and

outbound, must go through the Honeywall. Look into the figure below you can see that the Honeywall separates the production systems and the Honeynet itself. [42]

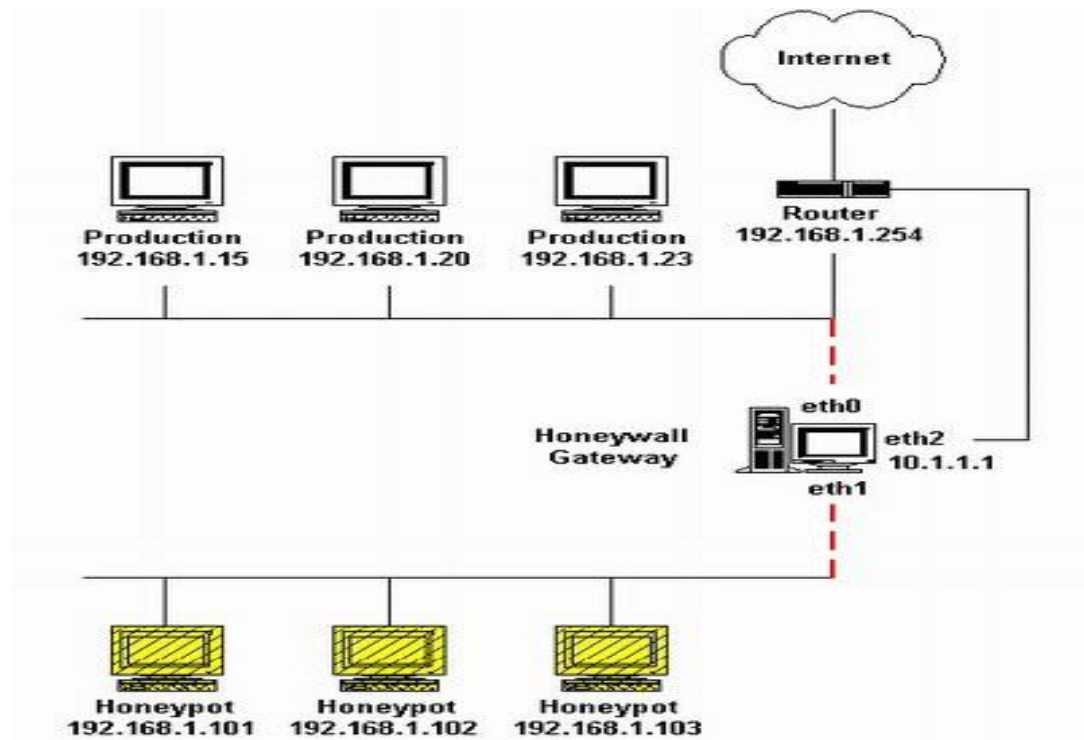


Fig10: 2st Generation Honeynet [41]

In conclusion, the Honeynet has four main functions: acquisition of data, control data, data collection and data analysis. However, one of the most serious problems of network administrators and security managers is the resource availability to manage the honeypot and the cost of implementing it. Small companies usually reject honeypots or honeynets simply because they do not have enough money to spend for it.

4.5 Padded cell

Padded cell is a kind of harden honeypot that attracting attackers with attracting data. A padded cell operates in parallel with a traditional intrusion detection/prevention system (IDPS).

When the IDPS detects attackers, it simply transfers them to a special virtual environment where they can cause no harm more, meaning it gives approach to the padded cell. Work same as honey pots, a padded cell has no useful purpose other than to catch attackers. It may be used as a form of an IDS. While it sounds fun and interesting, they need to be maintained and monitored: if an attacker does gain entry, they now can attack from within the network.

5. Some concerns of information security and the future of IDS

5.1 CIA triad of information security

Most computer attacks damage a system's security in many various ways. For example, an attack based on security hole may enable a hacker to access peculiar files or take over the admin role in system or allow access to a production network. Despite the diversified capabilities of computer attacks, they usually result in violation of only the CIA triad properties: availability, confidentiality, integrity.

The CIA triad of information security was a designed model to guide policies for information security within an organization despite of the underlying system and/or organization [43]:

Confidentiality: Ensures that data or an information system is hided from those people who are unauthorized person. Cryptography, encryption, passwords, access control lists (ACL) and policy based security are some of the methods through which confidentiality is achieved

Integrity: Integrity assures that the data or information system is authentic and unaltered from original information. One type of security attack is to sniff some important data and make changes to it before sending it on to the target receiver. Data encryption and hashing algorithms are key processes in providing integrity

Availability: Data and information systems are readily accessible when needed. Some types of security attack attempt to deny access to the authorized user. Hardware maintenance, software patching/upgrading and network optimization ensures availability

5.2 Comparing IDS with firewall

A firewall is a hardware and/or software which works in a networked environment to intercept unaccredited access while allowed authorized communications. Firewall is a device and/or a software that stands between an internal local network and the Internet, and filters traffic that might be harmful.

An Intrusion Detection System (IDS) is a software or hardware device which is set-up on the network (NIDS) or host (HIDS) to check and record intrusion attempts to the network.

I suppose that a firewall as security guard at the gate and an IDS device is a security camera after the gate. A firewall can restrict connection, while an Intrusion Detection System (IDS) cannot stop connection. It only notifies any intrusion attempts to the security administrator while an Intrusion Detection and Prevention System (IDPS) can block connections if it finds the connections is an intrusion attempt. [44]

5.3 Types of Computer Attacks Commonly Detected by IDSs

Three types of computer attacks are most widely detected by IDSs (Rebecca Bace and Peter Mell,2001) [45]: system scanning, denial of service (DOS), and system penetration. These attacks can be from internal network or remotely, using a network to access the target. An IDS operator must observe these attack types then react accordant to them

5.3.1 Scanning Attacks

A scanning attack occurs when an attacker explores a target network or system by sending different kinds of packets to looking for vulnerability or

weaknesses that can be exploited. When receiving response packets from the target system, they are analyzed to determine where are the vulnerabilities located. Application scanners, port scanners, vulnerability scanners, which are described on chapter 4 are used for this purpose to get these information (illustrated in Figure 11 below)

Scanning is typically considered a legal activity and there are a number of examples of legit scanning. The most well-known example of scanning applications are Web search engines or any netizen scan a network or the entire Internet looking for certain information, such as a music or video file. Some well-known malicious scanning including port scanning, ping scanning, very slow scan, scanning from multiple ports and scanning of multiple IP addresses and ports. NIDS signatures can be planned to identify such malicious scanning activity from a legitimate scanning activity with fairly high degree of accuracy

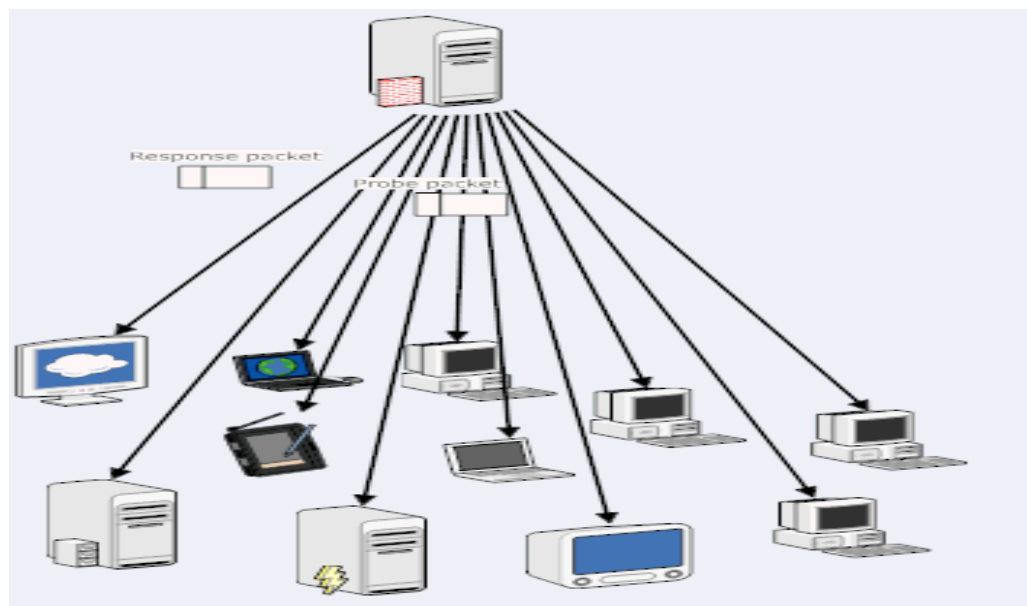


Fig11: Diagram shows a scanning attack where a single attack host scans a number of victims [46]

5.3.2 Denial of Service (DoS) Attacks

Denial of Service (DOS) attacks attempt to deny the legitimate and authorized users can access by slowing or shut down targeted network systems

or services. In certain Internet communities, DOS attacks are popular where a collection of bots are often used to attack web servers with dummy requests (illustrated in Figure 12). Such attacks can cause significant economic damage to electronic commerce operations by denying the customers a make purchases to the business.

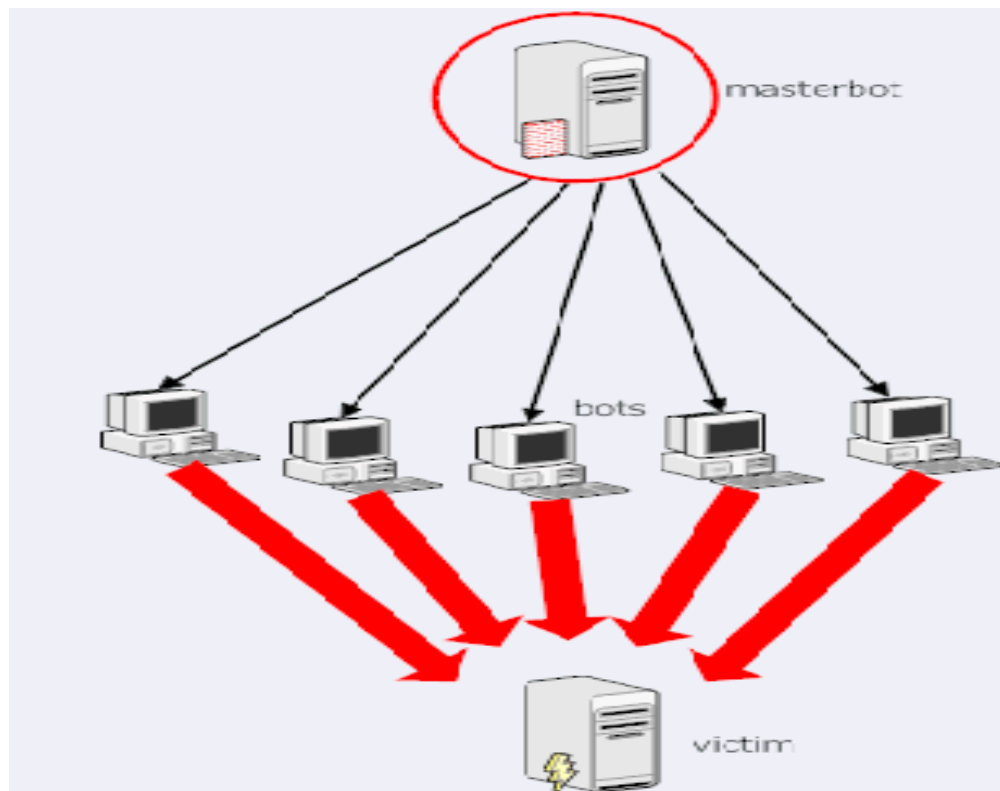


Fig12: Diagram shows a denial of service attack (DDoS in this case) [46]

There are three main of different kinds of DDoS attacks: application layer attacks, protocol attacks and volume based attacks. It is important for an IDS operator to understand the difference between them.

- Application layer attacks: Includes low-and-slow attacks, GET/POST floods, attacks that target Apache, Windows or OpenBSD vulnerabilities and more. Comprised of seemingly legitimate and innocent requests, the goal of these attacks is to bombard the web server, and the size is measured in requests per second. [47]

- **Protocol Attacks:** Includes SYN floods, fragmented packet attacks, Ping of Death, Smurf DDoS and more. This type of attack consumes actual server resources, or those of intermediate communication equipment, such as firewalls and load balancers, and is measured in packets per second. [47]
- **Volume Based Attacks:** Includes UDP floods, ICMP floods, and other spoofed-packet floods. The attack's goal is to saturate the bandwidth of the attacked site, and magnitude is measured in bits per second (Bps). [47]

5.3.3 Penetration Attacks

In penetration attack, a hacker gains an unauthorized control of a system involving to load a Trojan horse into the target systems. These malicious Trojan can be delivered either through some input medium (e.g., USB or CD-ROM), download, or e-mail; or by exploiting existing bugs and security flaws in such programs as Internet browsers. Activation need not be deliberate (e.g., double-clicking an icon), but can also occur by executing compromised code that users intentionally download from the Internet (e.g., device drivers, browser plug-ins, and applications) or even the simple viewing of a message in the preview screen of an e-mail client has, in some cases, proved sufficient to trigger execution of its attachment. [48]

5.3.4 The Future of IDSs

One of the key challenges with NIDS has basically should be the limitation on performance due to network growth. Currently, we see the growth up of network traffic of system network more than over last decade. So, IDS meets the trouble of capture all the packets, analyze the stream and process it in the a timely manner. Current systems can easily scale out to gigabit throughputs, and in future performance is likely to become less of an issue.

A more challenging problem is the reaction once an alarm has been raised. The center issue here is that this reaction is performed by human, and no matter how skilled the NIDS analyst is, this operation will remain slow and error prone. Recently, this step is going to be automated by some modern intrusion prevention system, which not only detects an attack but also takes an appropriate action upon certain attacks. [46]

The future of IDS lies in encryption data. The future of IDS is surely promising; however, it is important to view that how easy to spoof IP addresses, any source IP address with many tools are freely available on the Internet. And the future of generally encrypted traffic is rapidly approaching. IPv6 and large wireless networks signify for the trouble analysis used by most NIDS. The signatures which we are depending on would help much longer if they can't be matched against any data.

The big challenge then remains is the producing some algorithms that can detect anomalies with a quite fairly high degree of trust. Although many corporations give moneys on this research topic, it still is uncertain when such algorithms will be out of market and can be used in a commercial setting. [46]

In conclusion, the next generation of IDS should be deployed on cloud, having real time analysis and focus on broader context, beyond detection of malware and exploits only.

6. Snort

6.1 What is Snort

Snort is de facto standard technology worldwide open source network intrusion prevention and detection system. It uses a rule-based language combining signature, protocol and anomaly inspection methods. Currently, Snort is the most widely deployed intrusion detection and prevention technology. [49]

It can be connected to the most important databases such as Oracle, MySQL... It is constituted of an attack detection engine plus a port scanner, which allows alerting or responding to any kind of pre-defined attack. Snort also connect with some software to provide a simple GUI interfaces such as IDScenter or a web application like ACID or BASE that will get data from the database and will show it in a friendlier html format. However, these tools are old fashion and not updated anymore from their website.

Snort has an easy rule creation language and powerful in which several packs of rule-packages can protest against DoS, Nmap, backdoors... that can be downloaded from the snort webpage and many other sites. Moreover, Snort can work as a sniffer like Ethereal so the traffic in the network can be shown. Snort can be installed on Linux, Unix and Windows. It has a database of known patterns as well as new updates every time a new generation of attack has been founded.

6.2 Snort Architecture

Firstly, Snort is a kind of packet sniffer software such as popular Wireshark. Snort is developed gradually to become an IDS as seen today. It includes five main elements up to now: detection engines, logging and alert system, output plugin, decoder, pre-processor.

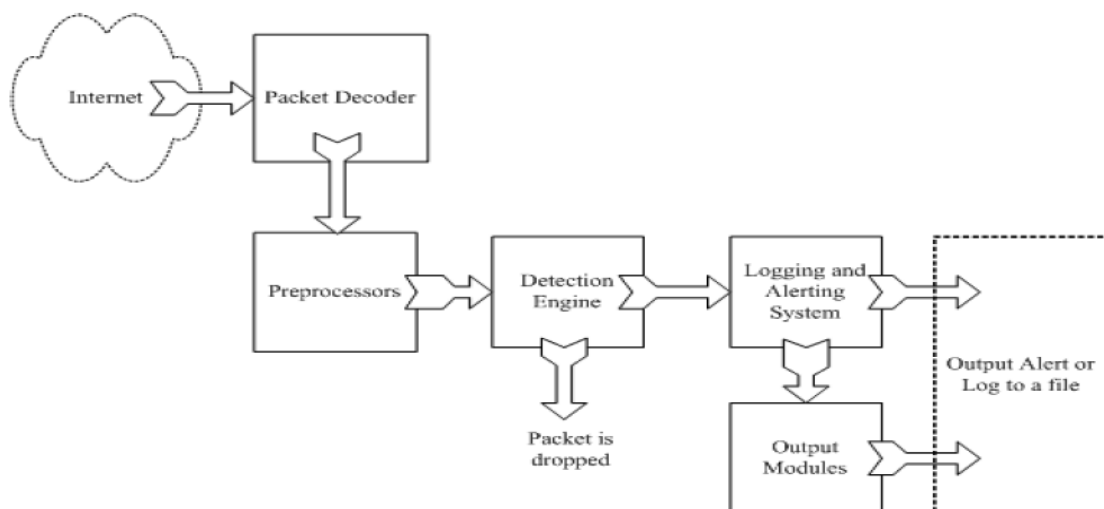


Fig13: Components of Snort [50]

1. The packet decoder – it collects network traffic from very different types of network interfaces (Ethernet, PPP...) and the raw data (packets) are being sent to the preprocessors.

2. The preprocessors is a plug-in which used by Snort to perform certain actions to modify raw data to scope out if the packet is used by intruder. Preprocessors are important for any IDS to prepare data packets to be studied base on rules in the detection engine. Preprocessors in Snort can defragment packets, decode HTTP URI, re-assemble TCP streams and so on. Having defined protocols, scan types and sensitivity levels it can identify multiple packets as a port scan. After doing its job, the processors will send the information to the detection engine.

3. The detection engine measures each data packet with each rule from a predefined ruleset. If packets equal the rule contents, they are being sent to the output. The detection engine is the time-consuming part of Snort. Amount of times would depend on the powerful of your machine and how many rules you have defined to respond to different packets. If traffic volume on your network is too high when Snort is working in NIDS mode, some packets maybe dropped and may not get a true real-time response.
[50]

4. Logging and Alerting System: After the detection phase, the checked packet may be used to note the activity or produce an alert. Logs are saved in simple text files, normally tcpdump type files.

5. The output – will give notifications or trigger alerts based on the rule action. Moreover, logs and alerts are a bit difficult to read from the command line – so this is why we need some user interfaces here. Snort user interfaces (Snorby, ACID) are implemented as the Output component of Snort. Notice that some tools are quite old and outdated at the moment.

6.3 Installation and configuration

6.3.1 Installation Snort

Snort often installed on a Linux machine, however Snort also offer to setup on a window machine. It's really headache when set it up on Windows compared to Linux environment because of lacking document guiding it on Windows. This is the explanation how we install and configure Snort on a computer running Windows.

The first tool we needed is WinCap [51] which Snort uses this library to sniff all the packets from our NIC. Installation of WinPCap is pretty easy. For our installation, we use the 4.1.3 version and accepting all of the default settings.

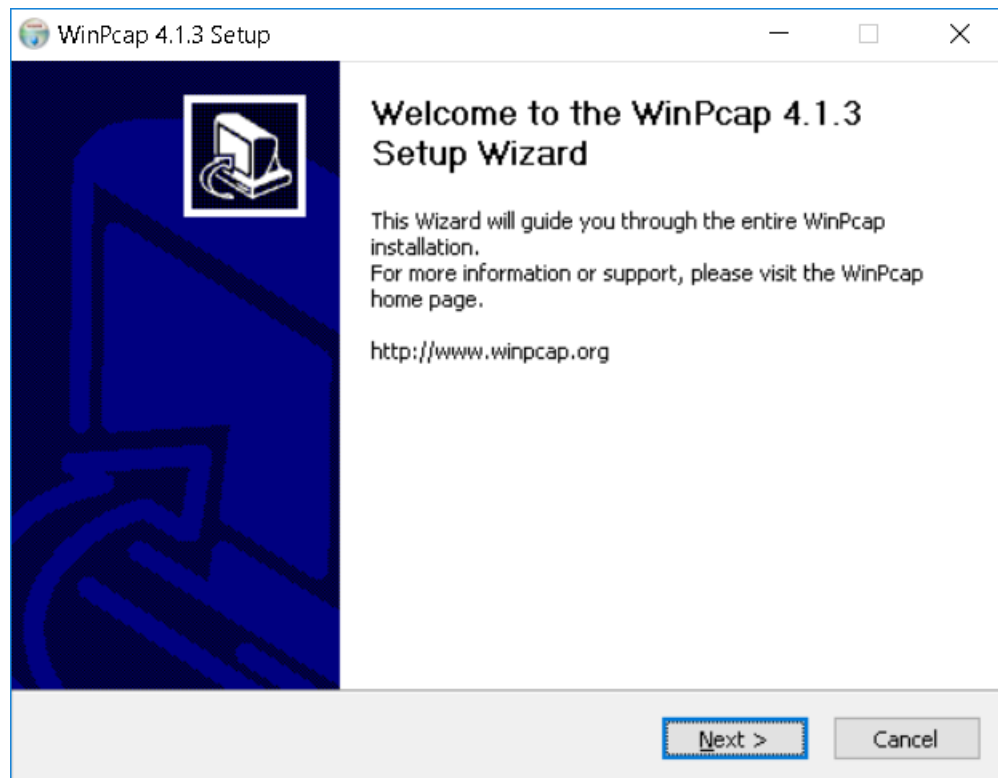
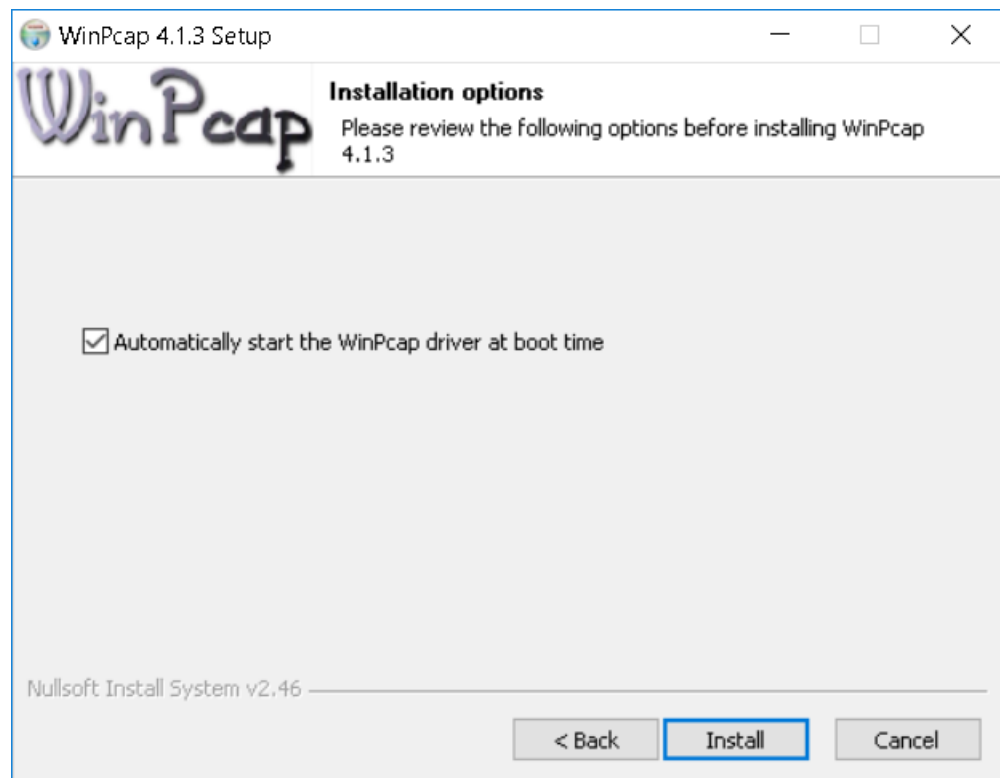
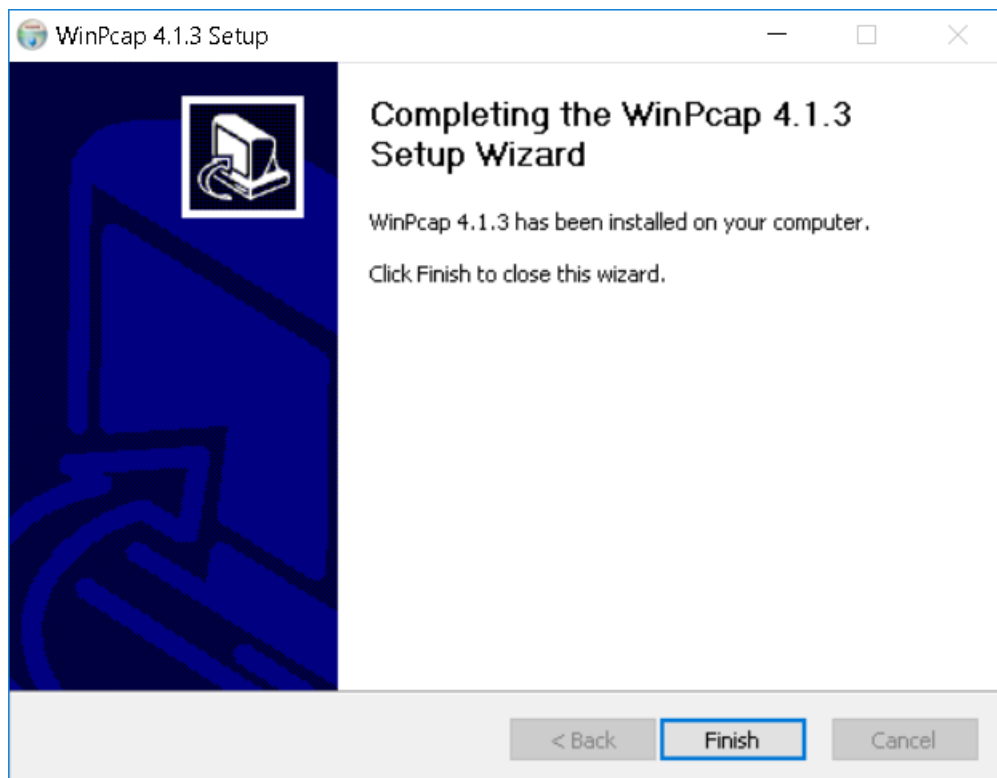


Fig14: Installation of WinPcap



Click on Install button



From the Snort website [52], download the file for windows and proceed to install it. I get the 2.8.0.1 as from version 2.9.3.0 Snort has no longer support database in configuration file. In this thesis, I'm trying to demonstrate how to connect the MySQL database working with Snort on Windows platform, and Linux based platform is out-of-scope. To install, we double click on Snort exe file to install it as followed.

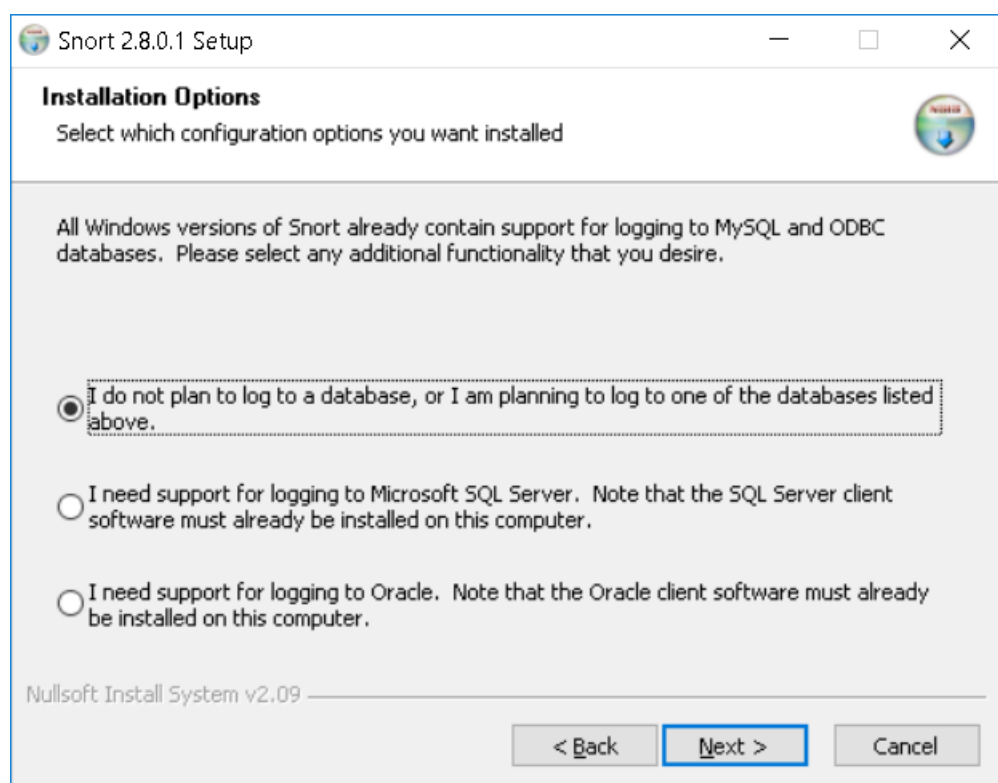
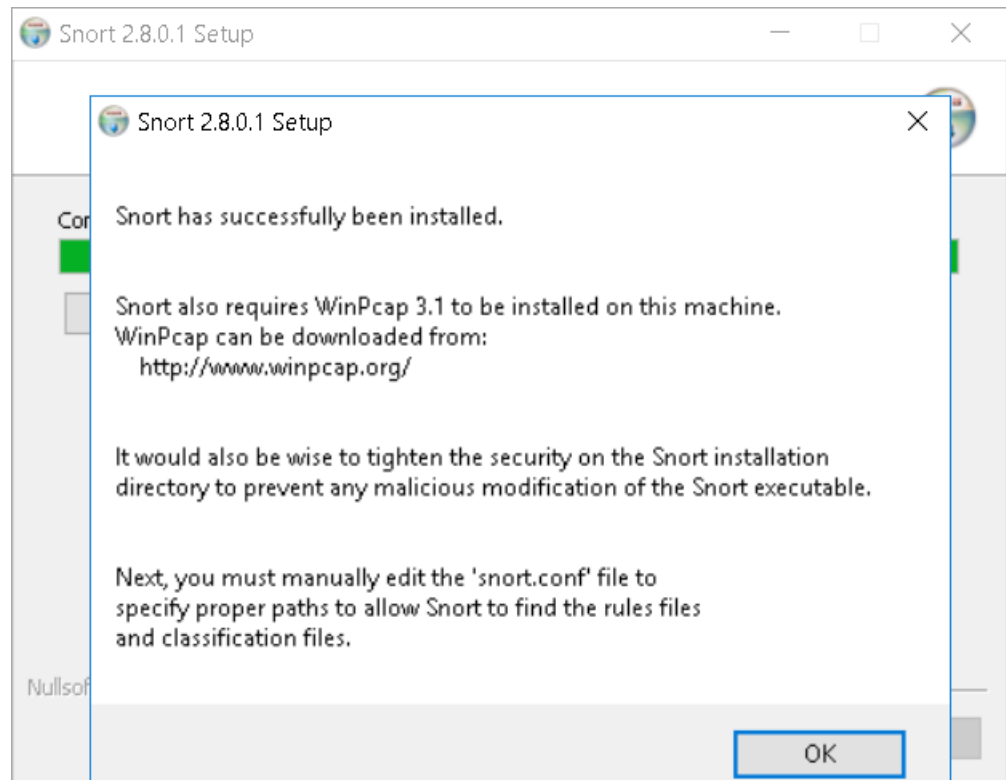
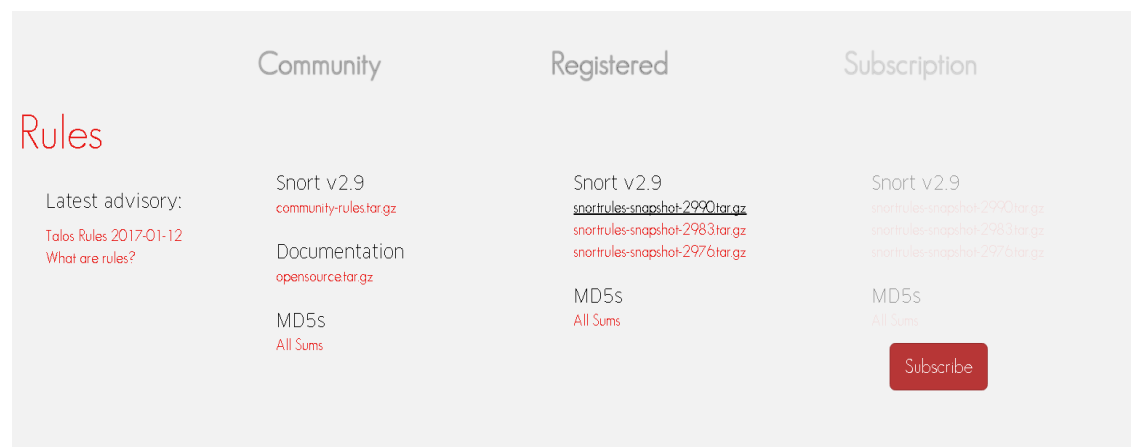


Fig15: Installation of Snort



Next, download snort rules on Snort website [52] and extract it into "C:\Snort\rules" folder (we download the registered version, note that we can use the latest rules for earlier snort version).



6.3.2 Configuration Snort

As noticed when finishing setup Snort application, we will need to change a couple of parameters in the c:\snort\etc\snort.conf file. To do so, let's use

Notepad++ application [53], WordPad, Notepad or whatever you like. Open the snort.conf file on C:\Snort\etc and configure 6 parts as followed:

- 1- Network variables
- 2- Dynamic libraries
- 3- Preprocessors (not modified, let it as default)
- 4- Output plugins
- 5- Configuration directives
- 6- Rule set

Note: Commenting with # will ignore the line when executing

Step #1: Set the network variables

You can set the IP of your need-protected host or set any

```
var HOME_NET 192.168.1.3/24
```

```
var HOME_NET any
```

Step #2: Change path of the "dynamicengine" variable value in the

"snort.conf" file with the path of your system as depict below:

```
"
dynamicpreprocessor directory c:\Snort\lib\snort_dynamicpreprocessor
#
# Load a specific dynamic preprocessor library from the install path
# (same as command line option --dynamic-preprocessor-lib)
#
# dynamicpreprocessor file /usr/local/lib/snort_dynamicpreprocessor/libdynamicexample.so
#
# Load a dynamic engine from the install path
# (same as command line option --dynamic-engine-lib)
#
dynamicengine c:\Snort\lib\snort_dynamicengine\sf_engine.dll
"
```

Step #3: Pre-processors configuration is not modified, let it's be as default.

Step #4: Configure output plugins

It will set which are the outputs that Snort will use when it produces alerts. Snort can store the data in many ways: to a simple .log file, databases such as MySQL, Oracle, SQL Server... and even can send the alert to a remote machine using the Syslog service.

We set the .log file storage with the fast option that will show the result in just one line with the most important details of the attack.

```
output alert_fast: snort.log
```

Or in my case, to set it work with MySQL , input these lines (notice that "snort" is database's name):

```
output database: alert, mysql, user=root password=root dbname=snort
host=localhost
```

```
output database: log, mysql, user=root password=root dbname=snort
host=localhost
```

Step #5: Add complete path for "include classification.config" and "includeeference.config" files

```
# metadata reference data. do not modify these lines
include C:\Snort\etc\classification.config
include C:\Snort\etc\reference.config
```

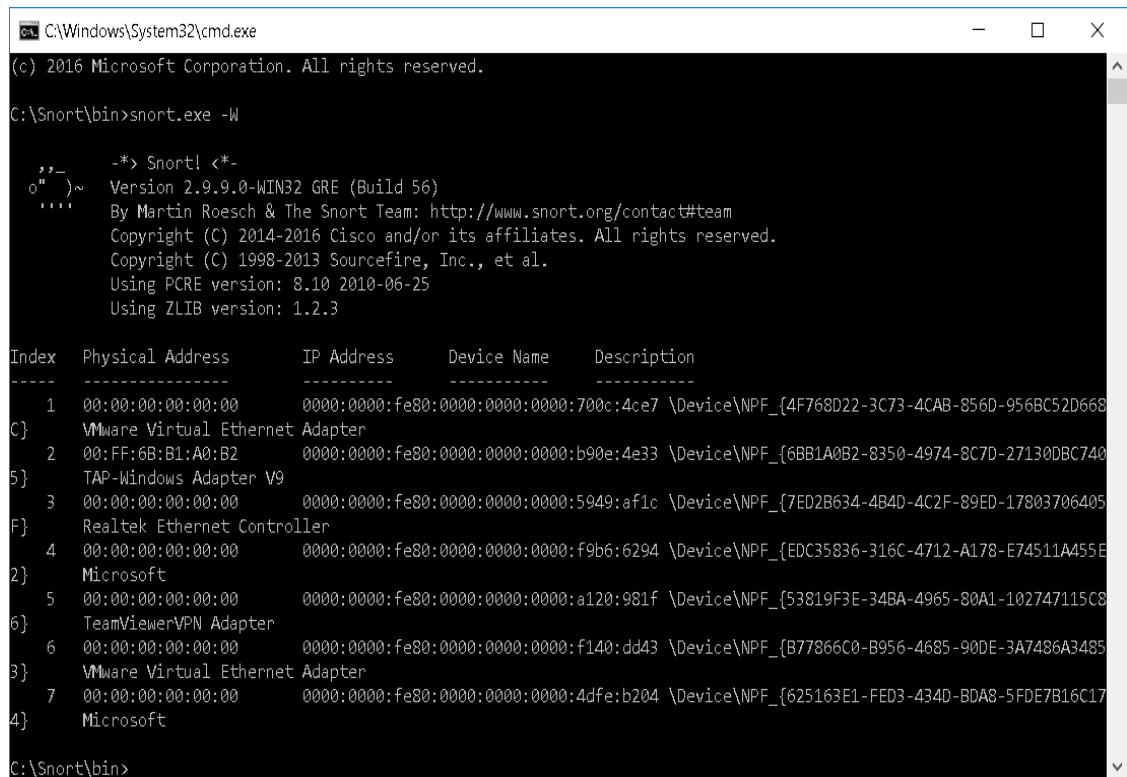
Step #6: Customize your rule set

The most common rules are already included for us here, or you can comment out any rules you don't want to include by insert # before these lines.

```
include $RULE_PATH/local.rules
include $RULE_PATH/bad-traffic.rules
include $RULE_PATH/exploit.rules
include $RULE_PATH/scan.rules
include $RULE_PATH/finger.rules
include $RULE_PATH/ftp.rules
include $RULE_PATH/telnet.rules
include $RULE_PATH/rpc.rules
include $RULE_PATH/rservices.rules
include $RULE_PATH/dos.rules
include $RULE_PATH/ddos.rules
include $RULE_PATH/dns.rules
include $RULE_PATH/tftp.rules

include $RULE_PATH/web-cgi.rules
```

We go to folder C:\Snort\bin, execute command snort.exe -W to check machine's interfaces



```

C:\Windows\System32\cmd.exe
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Snort\bin>snort.exe -W

,,_  -*> Snort! <*-
o"  )~ Version 2.9.9.0-WIN32 GRE (Build 56)
**** By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
      Copyright (C) 2014-2016 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

Index  Physical Address      IP Address      Device Name      Description
-----
1      00:00:00:00:00:00      0000:0000:fe80:0000:0000:0000:700c:4ce7 \Device\NPF_{4F768D22-3C73-4CAB-856D-956BC52D668
C} VMware Virtual Ethernet Adapter
2      00:FF:6B:B1:A0:B2      0000:0000:fe80:0000:0000:0000:b90e:4e33 \Device\NPF_{6BB1A0B2-8350-4974-8C7D-27130DBC740
5} TAP-Windows Adapter V9
3      00:00:00:00:00:00      0000:0000:fe80:0000:0000:0000:5949:af1c \Device\NPF_{7ED2B634-4B4D-4C2F-89ED-17803706405
F} Realtek Ethernet Controller
4      00:00:00:00:00:00      0000:0000:fe80:0000:0000:0000:f9b6:6294 \Device\NPF_{EDC35836-316C-4712-A178-E74511A455E
2} Microsoft
5      00:00:00:00:00:00      0000:0000:fe80:0000:0000:0000:a120:981f \Device\NPF_{53819F3E-34BA-4965-80A1-102747115C8
6} TeamViewerVPN Adapter
6      00:00:00:00:00:00      0000:0000:fe80:0000:0000:0000:f140:dd43 \Device\NPF_{B77866C0-B956-4685-90DE-3A7486A3485
3} VMware Virtual Ethernet Adapter
7      00:00:00:00:00:00      0000:0000:fe80:0000:0000:0000:4dfe:b204 \Device\NPF_{625163E1-FED3-434D-BDA8-5FDE7B16C17
4} Microsoft

C:\Snort\bin>
  
```

As you can see, the computer in the example has seven interfaces from 1 to 7. If we wanted to use Snort as a sniffer and watch all traffic on first interface, we could issue the command "snort.exe -i1 -v". This command would run Snort in verbose mode (-v) and have it listen on interface 1 (-i1). [54]

Run this command line to sniff for the third interface:

snort -c c:\snort\etc\snort.conf -l c:\snort\log -i 3 or use following command while running snort in IDS mode for generating log files in ASCII mode:
snort -A console -i3 -c c:\Snort\etc\snort.conf -l c:\Snort\log -K ascii

If you can see the image below, congratulation, now we have Snort running under Windows platform.

```

==== Initialization Complete ====

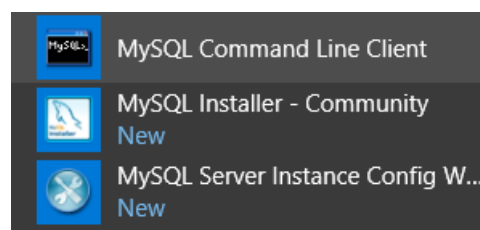
-*> Snort! <*-
Version 2.9.9.0-WIN32 GRE (Build 56)
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2016 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using PCRE version: 8.10 2010-06-25
Using ZLIB version: 1.2.3

Rules Engine: SF_SNORT_DETECTION_ENGINE Version 3.0 <Build 1>
Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>
Preprocessor Object: SF_SSH Version 1.1 <Build 3>
Preprocessor Object: SF_SMTP Version 1.1 <Build 9>
Preprocessor Object: SF_SIP Version 1.1 <Build 1>
Preprocessor Object: SF_SDF Version 1.1 <Build 1>
Preprocessor Object: SF_REPUTATION Version 1.1 <Build 1>
Preprocessor Object: SF_POP Version 1.0 <Build 1>
Preprocessor Object: SF_MODBUS Version 1.1 <Build 1>
Preprocessor Object: SF_IMAP Version 1.0 <Build 1>
Preprocessor Object: SF_GTP Version 1.1 <Build 1>
Preprocessor Object: SF_FTPTELNET Version 1.2 <Build 13>
Preprocessor Object: SF_DNS Version 1.1 <Build 4>
Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>
Preprocessor Object: SF_DCERPC2 Version 1.0 <Build 3>
Commencing packet processing (pid=768)

```

6.3.3 Installing MySQL

From Snort 2.9.3.0, Snort was going to remove the spo_database output module as well as Aruba and Prelude outputs. So, we use Snort 2.8.0.1 to connect to MySQL on Window version. Actually, we find some difficulties when setup Snort on Windows due to it was really supported more than on Linux/Ubuntu platforms. First thing first, we download MySQL [55] and install it; I'm not mention how to install it here as you can google it on the internet or follow the guide [56]. After finish installing, launch MySQL Command Line Client (console).




```
C:\Snort\mysql\bin\mysql.exe
Could not open required defaults file: C:\Snort\mysql\my.ini
Fatal error in defaults handling. Program aborted
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 45
Server version: 5.1.49-community MySQL Community Server (GPL)

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL v2 license

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

1. Create a new database into which to log Snort events:

```
mysql> create database snort;
```

Note: every action inside mysql must be finished with character “;”

2. Together with Snort installation, we have been provided with SQL scripts to create tables. The SQL scripts are in %Snort%\schemas subfolder.

This PC > Local Disk (C:) > Snort > mysql > schemas

Name	Date modified	Type	Size
create_db2	7/6/2007 10:57 PM	File	8 KB
create_mssql	7/6/2007 10:57 PM	File	10 KB
create_mysql	7/6/2007 10:57 PM	File	9 KB
create_oracle.sql	7/6/2007 10:57 PM	SQL File	10 KB
create_postgresql	7/6/2007 10:57 PM	File	8 KB
Makefile.in	11/14/2007 9:32 PM	IN File	10 KB

```
mysql> mysql -D snort -u root -p < c:\snort\mysql\schemas\create_mysql
```

3. Create a dedicated database user that Snort will use to log to the database.


Note: All the examples below assume that the database name is "snort", and MySQL users consist of a user name and a hostname

```
mysql> grant INSERT,SELECT,UPDATE,CREATE,DELETE,EXECUTE on
snort.* to snort@snort.host;
```

4. To test if database is ok, we use commands below to check again

```
mysql> use snort;
```

```
mysql> show tables;
```



```
C:\Snort\mysql\bin\mysql.exe
and you are welcome to modify and redistribute it under the GPL v2 license

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use snort;
Database changed
mysql> show tables;
+-----+
| Tables_in_snort |
+-----+
| data             |
| detail           |
| encoding          |
| event            |
| icmp_hdr         |
| ip_hdr           |
| opt              |
| reference         |
| reference_system |
| schema           |
| sensor           |
| sig_class        |
| sig_reference    |
| signature        |
| tcp_hdr          |
| udp_hdr          |
+-----+
16 rows in set (0.31 sec)

mysql>
```

6.3.4 Testing attacks

6.3.4.1 Test with Nmap attack:

Nmap ("Network Mapper") is a free and open source (license) utility for port scanning of the targeted machine as well as many other options such as getting the OS, what kind of firewalls are in use and services on open ports. [57]

This tool obviously can be used in a bad way as I'm attend to do it now to attack our host target PC. In the following figure is shown some of the captures of the program when performing the "attack":

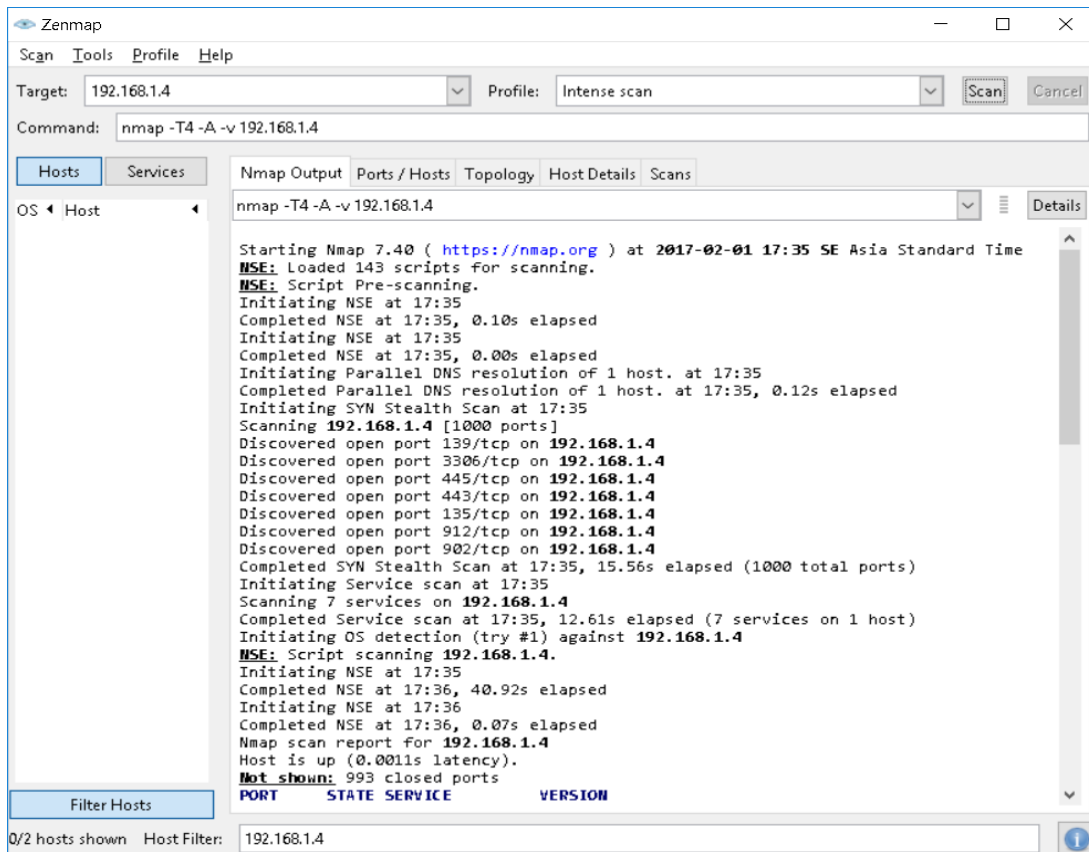


Fig16: ZenMap tool

Then run this following command while running snort in IDS mode for generating log files in ASCII mode:

snort -A console -i3 -c c:\Snort\etc\snort.conf -l c:\Snort\log -K ascii. Then Snort will detect the scan port progressing:

```

C:\Windows\System32\cmd.exe - snort -A console -i2 -c c:\Snort\etc\snort.conf -l c:\Snort\log -K ascii

--== Initialization Complete ==--

_*> Snort! <*-
o" )~
'...'
Version 2.8.0.1-ODBC-MySQL-FlexRESP-WIN32 (Build 72)
By Martin Roesch & The Snort Team: http://www.snort.org/team.html
(C) Copyright 1998-2007 Sourcefire Inc., et al.
Using PCRE version: 7.4 2007-09-21

Rules Engine: SF_SNORT_DETECTION_ENGINE Version 1.6 <Build 11>
Preprocessor Object: SF_SSH Version 1.0 <Build 1>
Preprocessor Object: SF_SMTP Version 1.0 <Build 7>
Preprocessor Object: SF_FTPTELNET Version 1.0 <Build 10>
Preprocessor Object: SF_DNS Version 1.0 <Build 2>
Preprocessor Object: SF_DCERPC Version 1.0 <Build 4>
Not Using PCAP_FRAMES
02/01-17:29:05.458690  [**] [122:3:0] (portscan) TCP PortswEEP [**] [Priority: 3] {PROTO:255} 192.168.1.4 ->
02/01-17:29:05.458690  [**] [122:1:0] (portscan) TCP Portscan [**] [Priority: 3] {PROTO:255} 192.168.1.4 ->
02/01-17:29:22.366741  [**] [122:3:0] (portscan) TCP PortswEEP [**] [Priority: 3] {PROTO:255} 192.168.1.4 ->
02/01-17:30:06.519180  [**] [122:1:0] (portscan) TCP Portscan [**] [Priority: 3] {PROTO:255} 192.168.1.4 ->
02/01-17:30:22.630978  [**] [122:3:0] (portscan) TCP PortswEEP [**] [Priority: 3] {PROTO:255} 192.168.1.4 ->
02/01-17:31:01.767057  [**] [119:4:1] (http_inspect) BARE BYTE UNICODE ENCODING [**] [Priority: 3] {TCP} 192
83 -> 192.168.1.1:80
02/01-17:31:02.054321  [**] [119:4:1] (http_inspect) BARE BYTE UNICODE ENCODING [**] [Priority: 3] {TCP} 192
89 -> 192.168.1.1:80
02/01-17:31:02.221055  [**] [119:4:1] (http_inspect) BARE BYTE UNICODE ENCODING [**] [Priority: 3] {TCP} 192
92 -> 192.168.1.1:80
  
```

6.3.4.2 Test with Death ping:

In a Ping of Death attack, the attacker sends a fragmented PING request that exceeds the maximum IP packet size (64KB). It's just a test for Snort because it's quite a very simple DDOS for a newbie.

Go to folder C:\Snort\rules, open file local.rules, add a new rule to detect is there someone try to ping our host:

```
alert icmp any any -> $HOME_NET any (msg: " Someone has ping you "; sid: 140791;)
```

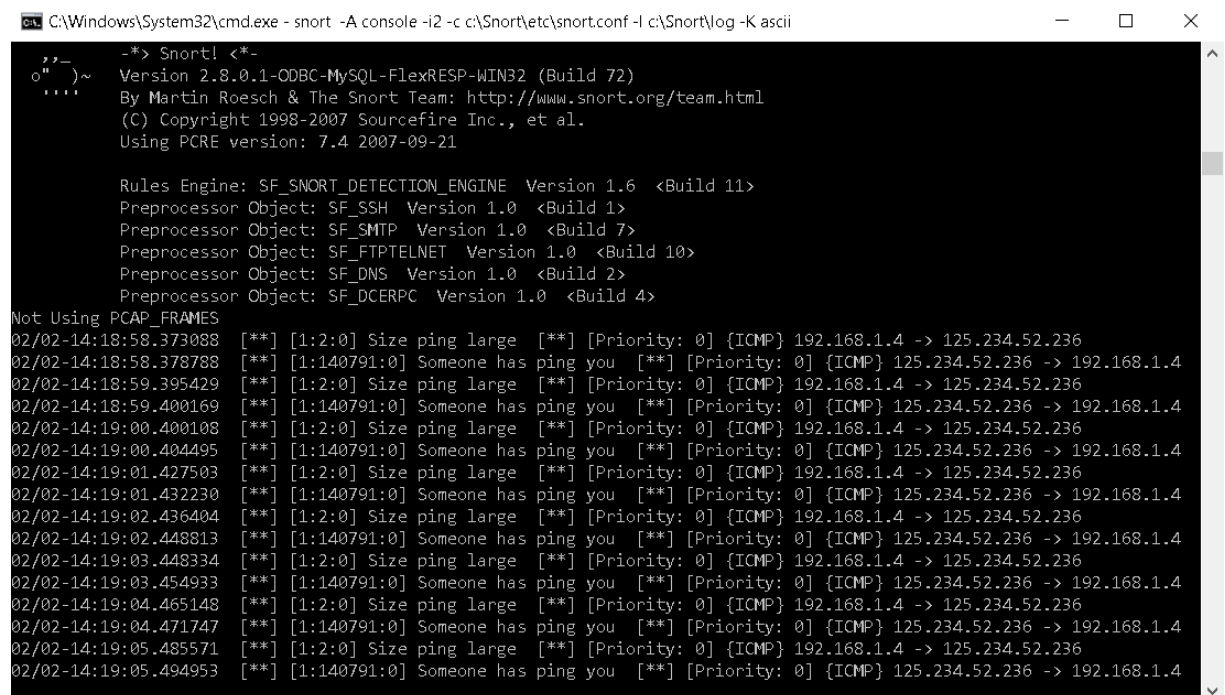
Open file icmp.rules, add a new rule to detect is there someone try to ping our host:

```
alert icmp $HOME_NET any -> any any (msg: " Size ping large "; dsize: >50; sid: 2;)
```

Then run this following command while running snort in IDS mode for generating log files in ASCII mode:

```
snort -A console -i3 -c c:\Snort\etc\snort.conf -l c:\Snort\log -K ascii
```

Open another window console and ping to your host, and Snort will detect the ping session



```

C:\Windows\System32\cmd.exe - snort -A console -i2 -c c:\Snort\etc\snort.conf -l c:\Snort\log -K ascii

-*> Snort! <*-
Version 2.8.0.1-ODBC-MYSQL-FlexRESP-WIN32 (Build 72)
By Martin Roesch & The Snort Team: http://www.snort.org/team.html
(C) Copyright 1998-2007 Sourcefire Inc., et al.
Using PCRE version: 7.4 2007-09-21

Rules Engine: SF_SNORT_DETECTION_ENGINE Version 1.6 <Build 11>
Preprocessor Object: SF_SSH Version 1.0 <Build 1>
Preprocessor Object: SF_SMTP Version 1.0 <Build 7>
Preprocessor Object: SF_FTPTELNET Version 1.0 <Build 10>
Preprocessor Object: SF_DNS Version 1.0 <Build 2>
Preprocessor Object: SF_DCERPC Version 1.0 <Build 4>
Not Using PCAP_FRAMES
02/02-14:18:58.373088 [**] [1:2:0] Size ping large [**] [Priority: 0] {ICMP} 192.168.1.4 -> 125.234.52.236
02/02-14:18:58.378788 [**] [1:140791:0] Someone has ping you [**] [Priority: 0] {ICMP} 125.234.52.236 -> 192.168.1.4
02/02-14:18:59.395429 [**] [1:2:0] Size ping large [**] [Priority: 0] {ICMP} 192.168.1.4 -> 125.234.52.236
02/02-14:18:59.400169 [**] [1:140791:0] Someone has ping you [**] [Priority: 0] {ICMP} 125.234.52.236 -> 192.168.1.4
02/02-14:19:00.400108 [**] [1:2:0] Size ping large [**] [Priority: 0] {ICMP} 192.168.1.4 -> 125.234.52.236
02/02-14:19:00.404495 [**] [1:140791:0] Someone has ping you [**] [Priority: 0] {ICMP} 125.234.52.236 -> 192.168.1.4
02/02-14:19:01.427503 [**] [1:2:0] Size ping large [**] [Priority: 0] {ICMP} 192.168.1.4 -> 125.234.52.236
02/02-14:19:01.432230 [**] [1:140791:0] Someone has ping you [**] [Priority: 0] {ICMP} 125.234.52.236 -> 192.168.1.4
02/02-14:19:02.436404 [**] [1:2:0] Size ping large [**] [Priority: 0] {ICMP} 192.168.1.4 -> 125.234.52.236
02/02-14:19:02.448813 [**] [1:140791:0] Someone has ping you [**] [Priority: 0] {ICMP} 125.234.52.236 -> 192.168.1.4
02/02-14:19:03.448334 [**] [1:2:0] Size ping large [**] [Priority: 0] {ICMP} 192.168.1.4 -> 125.234.52.236
02/02-14:19:03.454933 [**] [1:140791:0] Someone has ping you [**] [Priority: 0] {ICMP} 125.234.52.236 -> 192.168.1.4
02/02-14:19:04.465148 [**] [1:2:0] Size ping large [**] [Priority: 0] {ICMP} 192.168.1.4 -> 125.234.52.236
02/02-14:19:04.471747 [**] [1:140791:0] Someone has ping you [**] [Priority: 0] {ICMP} 125.234.52.236 -> 192.168.1.4
02/02-14:19:05.485571 [**] [1:2:0] Size ping large [**] [Priority: 0] {ICMP} 192.168.1.4 -> 125.234.52.236
02/02-14:19:05.494953 [**] [1:140791:0] Someone has ping you [**] [Priority: 0] {ICMP} 125.234.52.236 -> 192.168.1.4

```

6.3.4.3 Test with YouTube:

Go to folder C:\Snort\rules, add a new rule file name youtube.rules, and add a new rule to detect is there someone try to access to YouTube website:

```
alert tcp any any -> any any (content: "www.youtube.com"; msg: "You just visit youtube.com"; sid: 100000; rev: 1;)
```

In file snort.conf, add a line as: include \$RULE_PATH/youtube.rules
Then run this following command while running snort in IDS mode for generating log files in ASCII mode:

```
snort -A console -i3 -c c:\Snort\etc\snort.conf -l c:\Snort\log -K ascii
```

Open any browser and try to access YouTube website and Snort will detect the violation session

```
C:\Windows\System32\cmd.exe - snort -A console -i2 -c c:\Snort\etc\snort.conf -l c:\Snort\log -K ascii

--[AC-BNFA Search Info Summary]-----
| Instances      : 8
| Patterns       : 106
| Pattern Chars  : 698
| Num States     : 439
| Num Match States : 88
| Memory        : 18.97Kbytes
| Patterns       : 2.74K
| Match Lists    : 2.57K
| Transitions    : 12.98K
|-----|

---- Initialization Complete ----

--> Snort! <*-
o" )~ Version 2.8.0.1-ODBC-MySQL-FlexRESP-WIN32 (Build 72)
"    By Martin Roesch & The Snort Team: http://www.snort.org/team.html
(C) Copyright 1998-2007 Sourcefire Inc., et al.
Using PCRE version: 7.4 2007-09-21

Rules Engine: SF_SNORT_DETECTION_ENGINE Version 1.6 <Build 11>
Preprocessor Object: SF_SSH Version 1.0 <Build 1>
Preprocessor Object: SF_SMTP Version 1.0 <Build 7>
Preprocessor Object: SF_FTPTELNET Version 1.0 <Build 10>
Preprocessor Object: SF_DNS Version 1.0 <Build 2>
Preprocessor Object: SF_DCERPC Version 1.0 <Build 4>
Not Using PCAP_FRAMES
02/02-14:26:49.391732  [**] [1:100000:1] You just visit youtube.com [**] [Priority: 0] {TCP} 192.168.1.4:52618 -> 125.23
4.52.39:443
```

6.4 Compare Snort and Suricata

For years, Snort has been the common software of open source Intrusion Detection/Prevention Systems (IDS/IPS). Its engine comprises the benefits of signatures, protocols, and anomaly-based inspection and has become the most widely deployed IDS/IPS in the world.

Suricata, a new and less extensive product developed by the Open Information Security Foundation (OISF), has recently emerged, and seems really hopeful. It is also based on signatures but integrates revolutionary techniques. This engine embeds a HTTP normalizer and parser (HTP library) that provides very advanced processing of HTTP streams, enabling the understanding of traffic on the 7th level of the OSI model. Though Suricata's architecture is dissimilar from Snort, it behaves the same way as Snort and can use the same signatures [58].

After research on many documents [59] [60] [61] [62] [63], we can conclude on some points of Snort compared with Suricata:

- Overview: Both Suricata and Snort are very capable intrusion detection systems.
- Threaded: Suricata's multi-threaded architecture requires more memory and CPU resources than Snort. The aggregate CPU use of Suricata is nearly double that of Snort, and Suricata uses over double the amount of RAM used by Snort. Snort is highly efficient in the scenario of moderate traffic with a single core processor (single thread)
- Network traffic: Suricata has the advantage that it can grow to accommodate increased network traffic without requiring multiple instances. Snort is lightweight and fast but limited in its ability to scale beyond 200-300 Mbps network bandwidth per instance. Recent versions of Snort support PF-RING and PCAP acceleration providing support for higher traffic. Suricata provides support for PF-Ring, AF packet, PCAP acceleration and NFLOG
- Rule-set: both Snort and Suricata can use the same rule set and read the rules in similar ways.
- Scalability: Snort can be successfully deployed on any network environment. Suricata is more focused on large scale networks. In a way, it could be considered as an extension of Snort for large networks.
- Flexibility and Usability: Snort and Suricata can run on various operating systems including Linux, Windows, and Mac OS X.
- Support Community: There are vast community of users, many support resources available online for Snort. Suricata is quite new and less support.

- **Accuracy:** Suricata has a higher accuracy rate than Snort due to Snort often ignores many dropped packets by using only single-thread
- **Financial matter:** Both are Open Source Engine

As a conclusion, Snort remains the current standard for IDS/IPS in production environments. It is strong, easily configurable and very well documented. However, Suricata is a coming out IDS/IPS that could revolution the detection techniques and Snort will certainly imitate some of these features (support of multi-threading) in future releases. [58]

6.5 Improve your custom Snort's rules

6.5.1 Snort Rule Structure

A Snort rule, is composed by the header (information about the traffic) and the options (contains some action to do on the packet). [64]

- Headers is composed by:
 - Action, Protocol, Source IP, Source Port, Direction Operator, Destination IP, Destination Port (Options)
- Options: Contains the messages and information essential for the decision of the alert. The different options are separated with the character “;”. There are four main categories of options:
 - **Meta-data:** Provides some information about the rule.
 - **Payload:** Refers to the “useful” data from the interior of a packet, usually is known the body of the data.
 - **Non-payload:** Looks data in other parts of the packet different of the body data.
 - **Post-detection:** This option is a trigger for a rule when this is activating.

6.5.2 What's a Snort bad rule and improvement:

Mr. Leon Ward, a senior security engineer of Sourcefire, gives definitions what rules cans call a bad rule [65]:

- A rule that cannot or will not perform the function it's made for. I.e. it won't catch the attack/event that the rule tries to find (False negative)
- A rule that catches the wrong stuff (False positive)
- A rule that has little meaning or value to the analyst (Junk alerts)
- A rule that wastes processing time and effort to achieve its goal (performance hog)

This is a simple example of a bad rule:

alert tcp any any -> any any (msg: "Packets Detected")

This rule asks Snort to look for all packets in all packets flowing out of the network to TCP on any ports. This rule is really a junk alert and lacking many important information for the admin to understand of what's happening on the network, and what it means if it generates an alert.

So, for writing efficient rules, David J. Bianco gives us some suggestion that I summary which suitable as follows: [66]

- Be as specific as possible in the header
 - ✓ Beware of the "any" keyword
 - ✓ Specify the protocol, IP addresses and ports
- Use "flow: established" for TCP sessions
- Body options are evaluated in order until match is unsuccessful, so list broad matches first
- Use content matches to comb out packets that can't match
- Always backup your rules in the local.rules file
- If snort doesn't restart after you add your new rule, check /var/log/messages (Linux) or Snort\log (window) for details
- When writing a complex rule, start small and build it piece-by-piece

7. Conclusion

In conclusion, Snort is a light open source program from which very good results can be obtained. We have found it quite difficult and outdated when installing on Window platform because it mainly supports for Linux based environment. Therefore, we suggest using it with earlier version of Windows if we want to use it properly or install it on Ubuntu server but this is out of scope of this thesis. It has been used not only a detection system but also prevention functions such as closing the connection in case of detecting an attack. However, we believe that it need to improved strongly because of some reasons. One is the amount of dropped packets due to the high speed of ADSL and cable but it works fine in our small testing environment.

Another important challenge to Snort is how it deals with encrypted traffic. Some Snort rules that detect unusual encrypted traffic as well as VPN, but the real ado here is lied deep inside those packets and decrypt them without having the key. Snort will not notice attacks carried out by an encrypted channel [34]. Evidently, everything will just fine if we had the key of the encryption. Snort should be placed just behind the tool that decrypts the message. Last thing is Snort for wireless card is quite hard to setup, you can use Snort to sniff wireless traffic with two routers. For more information, we can check the guide here for clearer instruction. [67]

We should admit that intrusion detection systems are not needed for all organizations. If an organization cannot afford money for a network specialist on observing attack responses, having an intrusion detection system will not provide any additional security. We should note that the Intrusion detection systems are not flawless, since new malicious codes or holes are constantly coming up every day. Those can be exploited by the hacker to overcome the security system.

Although an intrusion detection system is a good choice to keep us aware of what is really going on in our network in case of attacks, this choice is not useful at all if we do not take care to other simply aspects for security such as having peculiar passwords in our systems, well-config firewall settings, or an always

available back up.

While statistics say that there is remarkably number of companies have IDS in their security [68], for us, using of Snort on our company is ideally setting up for a small department of company firstly because of currently possible danger of disclosing data before trying on our main dedicated server. Snort can cover around 90% the demand of our company's needs as we revise our requirement again as below:

- ✓ It should be constantly updated, which is a key part of the deploy decision: yes
- ✓ It should be designed to merge smoothly into the current network system of the enterprise: yes
- ✓ It features to detect critical threats, perform behavioral analysis: yes
- ✓ It should be integrated with the firewall: no
- ✓ Capable of signature matching and behavior anomaly detection: yes
- ✓ Capable of detection on high speed analysis: quite yes and no, near-future supported

We should advocate to use Snort for our working environments but still keep watching on Suricata since this ID could quickly be growth fast in a very near future and more suitable for large scale networks.

8. References

- [1] <http://vietnamnews.vn/opinion/in-the-spotlight/300848/viet-nams-network-security-at-high-risk.html> [Online, access November 2016]
- [2] <http://www.firewallinformation.com/> [Online, access November 2016]
- [3] Karen Scarfone, The basics of network intrusion prevention systems, <http://searchsecurity.techtarget.com/feature/The-basics-of-network-intrusion-prevention-systems>, [Online, access November 2016]
- [4] Rafeeq Ur Rehman, Intrusion Detection Systems with Snort: Advanced IDS Techniques Using Snort, Apache, MySQL, PHP, and ACID, p.7, 2003
- [5] Xu, D. and P. Ning, 2008. Correlation Analysis of Intrusion Alerts. In: Intrusion Detection Systems (Advances in Information Security), Pietro, R.D.and L.V. Mancini (Eds.), ISBN: 0387772650, pp: 65-92
- [6] Maggi, F., M. Matteucci and S. Zanero, 2009. Reducing false positives in anomaly detectors through fuzzy alert aggregation. Inform. Fusion, Volume 10, issue 4.
- [7] James P. Anderson, "Computer Security threat monitoring and surveillance", 1980
- [8] D. E. Denning, "An intrusion detection model." IEEE Transactions on Software Engineering, Feb. 1987

- [9] The Evolution of Intrusion Detection Systems by Paul Innella, Tetrad Digital Integrity, LLC <http://www.securityfocus.com/infocus/1514> (visited November 2016)
- [10] Martin Roesch: “Snort Documents”, <http://www.snort.org/docs/>
- [11] Anomaly based Network Intrusion Detection System by Dinakara K, p. 8
- [12] Intrusion Detection Systems (IDS) Part I - (network intrusions; attack symptoms; IDS tasks; and IDS architecture) by Przemyslaw Kazienko & Piotr Dorosz [Published on 7 April 2003 / Last Updated on 7 April 2003], from http://www.windowsecurity.com/articles-tutorials/intrusion_detection/Intrusion_Detection_Systems_IDS_Part_I_network_intrusions_attack_symptoms_IDS_tasks_and_IDS_architecture.html (visited November 2016)
- [13] H. Debar, M. Dacier, A. Wespi, Towards a taxonomy of intrusion-detection systems, Computer Networks 31, 1999, pages 805-822.
- [14] E. Lundin, E. Jonsson, Survey of research in the intrusion detection area, Technical report 02-04, Department of Computer Engineering, Chalmers University of Technology, Göteborg January 2002, http://www.ce.chalmers.se/staff/emilie/papers/Lundin_survey02.pdf.
- [15] C. Krügel, T. Toth, Applying Mobile Agent Technology to Intrusion Detection, ICSE Workshop on Software Engineering and Mobility, Toronto May 2001, from

- <http://www.elet.polimi.it/Users/DEI/Sections/Compeng/GianPietro.Picco/ICSE01mobility/papers/krugel.pdf>.
- [16] C. Krügel, T. Toth, Distributed Pattern Detection for Intrusion Detection, Conference Proceedings of the Network and Distributed System Security Symposium NDSS '02, 2002,
<http://www.isoc.org/isoc/conferences/ndss/02/proceedings/papers/kruege.ps>.
- [17] J.S. Balasubramanian, J.O. Garcia-Fernandez, D. Isaco, E. Spafford, D. Zamboni, An Architecture for Intrusion Detection using Autonomous Agents, 14th IEEE Computer Security Applications Conference ACSAC '98, December 1998, pages 13-24,
<http://www.cs.umbc.edu/cadip/docs/NetworkIntrusion/tr9805.ps>.
- [18] D.J. Ragsdale, C.A. Carver, J.W. Humphries, U.W. Pooh, Adaptation techniques for intrusion detection and intrusion response systems, Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 2000, pages 2344-2349,
<http://www.itoc.usma.edu/ragsdale/pubs/adapt.pdf>.
- [19] Abhishek Pharate, Harsha Bhat, Vaibhav Shilimkar, Classification of Intrusion Detection Systems, retrieve from
https://www.academia.edu/11395235/CLASSIFICATION_OF_INTRUSION_DETECTION_SYSTEMS, pages 1-6
- [20] Pengertian Intrusion Detection System (16 Jan 2016), retrieve

- from http://luwuk59.blogspot.com/2016/01/pengertian-intrusion-detection-system_16.html (visited November 2016)
- [21] Marion Bogdanov ,“An approach to developing an information assurance environment”,
https://www.academia.edu/10025339/An_Approach_to_Developing_An_Information_Assurance_Environment (visited November 2016)
- [22] Sailesh Kumar, Survey of Current Network Intrusion Detection Techniques, retrieve from <http://www.cse.wustl.edu/~jain/cse571-07/ftp/ids/> (visited November 2016)
- [23] Sanjay Kumar Sharma, Pankaj Pande, Susheel Kumar Tiwari and Mahendra Singh Sisodia, “An Improved Network Intrusion Detection technique based on k-means clustering via naïve Byes Classification”
- [24] Thanvarat Komviriyavut, Phurivit Sangkatsanee, Naruemon Wattanapongsakorn, “Network intrusion detection and classification with decision tree and rule based approach”
- [25] Deris Stiawan, Ala’ Yaseen Ibrahim Shakhatreh, Mohd. Yazid Idris, Kamarulnizam Abu Bakar, Abdul Hanan Abdullah, “Intrusion prevention system: a survey”.
- [26] N. Wattanapongsakorn, S. Srakaew, E. Wonghirunsombat, C. Sribavonmongkol, T. Junhom, P. Jongsubsook, C. Charnsripinyo, “A Practical Network based Intrusion Detection and Prevention System”

- [27] K.B.Chandradeep, “A scheme for the design and implementation of a distributed ids”
- [28] Kjetil Haslum, Ajith Abraham and Svein Knapskog, “Fuzzy online risk assessment for distributed intrusion prediction and prevention systems”
- [29] Hakan Albag, “Network & agent based intrusion detection systems.”
- [30] Vinod Kumar, Dr. Om Prakash Sangwan, “Signature based intrusion detection system using Snort”.
- [31] V. Jyothsna, V. V. Rama Prasad, K. Munivara Prasad, “A review of anomaly based intrusion detection systems”.
- [32] Noonan, W. J. (2004). Hardening network infrastructures: Bulletproof your systems before they get hacked. New York: Osborne.
- [33] Security All the Way, retrieve from <https://infosecprimer.wordpress.com/2013/07/09/introducing-ids-and-ips/> (visited November 2016)
- [34] An Evaluation of current IDS, Master thesis performed in Information Coding by Ignacio Porres Ruiz And María del Mar Fernández de Ramón, p 32
- [35] Johan Nilsson, Vulnerability scanners, May 2006, p 31-38

[36] Marco de Vivo, Eddy Carrasco, Germinal Isern and Gabriela O. de Vivo, A review of port scanning techniques, 1999,
<http://portal.acm.org/citation.cfm?id=505737>

[37] John Wack, Miles Tracy, Murugiah Souppaya Guideline on Network Security Testing, 2003, NIST Special Publication 800-42,
www.iwar.org.uk/comsec/resources/netsec-testing/sp800-42.pdf

[38] Jay Beale, Haroon Meer, Roelof Temmingh, Charl Van Der Walt, Renaud Deraison, Nessus Network Auditing, 2004,
<http://dl.acm.org/citation.cfm?id=993973>

[39] Loras R. Even, Honey Pot Systems Explained July 2000, retrieve from <https://www.sans.org/security-resources/idfaq/what-is-a-honeypot/1/9/> (visited December 2016)

[40] Duy Long, Tìm hiểu về "Honeypot" và "honeynet", retrieve from <http://quantrimang.com/tim-hieu-ve-honeypot-va-honeynet-37896/> (visited December 2016)

[41] <http://www.honeynet.org/> (visited December 2016)

[42] <http://homes.cerias.purdue.edu/~kaw/research/honeynet/HoneynetTutorial/honeynet/gen2.html> (visited December 2016)

[43] <https://www.techopedia.com/definition/25830/cia-triad-of-information-security> (visited December 2016)

[44] <http://www.omnisecu.com/security/infrastructure-and-email->

- [security/difference-between-firewall-and-intrusion-detection-system.php](#) (visited December 2016)
- [45] Rebecca Bace and Peter Mell, Intrusion Detection Systems (2001),
<http://csrc.nist.gov/publications/nistpubs/800-31/sp800-31.pdf>
- [46] Sailesh Kumar, Survey of Current Network Intrusion Detection Techniques, <http://www.cse.wustl.edu/~jain/cse571-07/ftp/ids/#sec6.1> (visited December 2016)
- [47] Distributed Denial of Service Attack (DDoS) Definition (June 2011), retrieve from <http://www.hostglobal.tech/security/distributed-denial-of-service-attack-ddos-definition/> (visited December 2016)
- [48] Penetration attack, http://itlaw.wikia.com/wiki/Penetration_attack (visited December 2016)
- [49] Yue Jiang, Snort - a network intrusion prevention and detection system, www.csee.wvu.edu/~cukic/CS665/Snort.ppt
- [50] Rafeeq Ur Rehman, Intrusion Detection Systems with Snort: Advanced IDS Techniques Using Snort, Apache, MySQL, PHP, and ACID, p.12-16, 2003
- [51] <https://www.winpcap.org/install/> (February, 2017)
- [52] <https://www.snort.org/downloads> (February, 2017)
- [53] <https://notepad-plus-plus.org/download/v7.3.1.html> (February, 2017)

- [54] <https://www.sans.org/security-resources/idfaq/running-snort-under-windows/6/4> (February, 2017)
- [55] MySQL database <http://www.mysql.org> (February, 2017)
- [56] <http://www.mysqltutorial.org/install-mysql/> (February, 2017)
- [57] <https://nmap.org/> (February, 2017)
- [58] <https://www.aldeid.com/wiki/Suricata-vs-snort> (March, 2017)
- [59] Chintan Kacha¹ & Kirtee A. Shevade, Comparison of Different Intrusion Detection and Prevention Systems, December 2012
- [60] http://wiki.aanval.com/wiki/Snort_vs_Suricata (March, 2017)
- [61] <https://priyachalakkal.wordpress.com/2016/03/24/suricata-snort-bro/> (March, 2017)
- [62] David J. Day & Benjamin M. Burns, A Performance Analysis of Snort and Suricata Network Intrusion Detection and Prevention Engines, Feb 2011
- [63] Albin, Eugene, A comparative analysis of the Snort and Suricata intrusion-detection systems, Sep 2011
- [64] <http://opentodo.net/2012/10/snort-from-scratch-part-iii/> (access on March, 2017)
- [65] Leon Ward, Improving your custom Snort rules, November 2010,

<http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=B32D105C35982E4AC8A1DB5C73789566?doi=10.1.1.225.1866>

[66] David J. Bianco, EZ Snort Rules, Find the Truffles, Leave the Dirt,

www.vorant.com/files/EZ_Snort_Rules.pdf

[67] <https://wiki.archlinux.org/index.php/snort> (February, 2017)

[68] <https://idatalabs.com/tech/products/snort> (February, 2017)

Appendix A: Acronym

Abbreviations	Description
ACM	Association for Computing Machinery
AD	Anomaly Detection
DARPA	Defense Advanced Research Projects Agency
DDoS	Denial of service
FTP	File Transfer Protocol
HIDS	Host intrusion detection system
ICMP	Internet Control Message Protocol
IDS	Intrusion detection system
IDWG	Intrusion Detection Working Group
IEEE	Electrical and Electronics Engineers
KDD	Knowledge discovery and data mining
KMIE	K-means based on information entropy
LAN	Local Area Network
NDC	Network data collector
NDM	Network data mining
NDP	Network data processor processes
NIDS	Network intrusion detection system
SMTP	Simple Mail Transfer Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VRT	The Sourcefire Vulnerability Research Team, currently named Talos