Abertay University

The Reliability of Intrusion Detection SYSTEMS

Cameron Cottam – 1901441@abertay.ac.uk

Introduction to Security – CMP110

BSc Ethical Hacking Year 1

2020

Note that Information contained in this document is for educational purposes.

.

Abstract

This report offers an overview of how Intrusion Detection systems work, and the different types of features offered by IDS software on different operating systems work. The report will highlight two types of IDS software that are used to monitor a client's activity on a host-based level and network-based level.

This report shows how to trigger alerts using Intrusion Detection Software. The Intrusion Detection System software that was used was Snort [1] and OSSEC [2], which were installed on a Windows 7 and Ubuntu virtual machine respectively. This was done to provide a safe environment in testing the IDS software. Some of the tests were installing a rootkit and launching attacks by using a Kali Linux virtual machine.

The results section details what exactly the IDS software detects and displays the source of origin on the machine or on a network. To which Snort and OSSEC provide results that are important in helping an Administrator monitor their systems.

Keywords:

Intrusion Detection Systems (IDS), Host-based (HIDS), Network Intrusion (NIDS), Intrusion Prevention System (IPS), NMAP, Snort, Open Source HIDS SECurity (OSSEC)

.

1 CONTENTS

.

.

2 INTRODUCTION

BACKGROUND

Intrusion detection systems (IDS) have an important job in maintaining the security of information systems and ensuring that it's well protected from malicious software, such as malware, phishing, etc. IDS can be either an application or hardware that helps to track a network or a system's activity for malicious software or policy violations. A policy violation could happen when any files have been modified without an administrator's permission to do so.

When an IDS detects that malware or any suspicious activity is occurring, the administrator is alerted to the issue that their system may be compromised or that the IDS has detected suspicious activity over a network. IDS can only detect attacks that are occurring, to prevent attacks on a system an Intrusion Prevention system is needed for this process.[3]

Intrusion detection systems are a security measure to help protect the integrity of a system. If the computer's firewall is unable to detect any attacks, an IDS can act as the barrier to alert the administrator of suspicious activity.

IDS detection works by analyzing what happened during an intrusion and tries to identify if the computer has been misused. There are two types of detection, Network intrusion detection systems (NIDS) – which a system would analyses the incoming network data. – and, Host-based intrusion detection systems (HIDS) – a system that monitors the critical operating system files. An example of IDS programs that are HIDS and NIDS are Snort and Open Source HIDS SECurity (OSSEC) respectively.

Host-based IDS can detect intrusions by using different methods, such as file integrity checking and log monitoring. Network-based IDS detection works by monitoring incoming traffic by capturing packets [4]. There are benefits and drawbacks to both IDS types. For example, Network-based IDS will be faster in response to an intrusion than host-based IDS as they monitor network packets as they receive it in real-time.[5] As well, Network-based IDS do not need to modify any files that might have changed, this is because NIDS monitors segments being sent to the client and can decide to drop them.

A major security issue is encrypted traffic being sent, which could contain malicious software. However, host-based IDS are much easier to analyze decrypted data with the right encryption keys. But also are unable to detect attacks that could spread in a network.[6]

There are sublayers of IDS, which work based on specific patterns and adapt to unknown attacks which are known as Signature-based and Anomaly-based. Signature-based looks at a database that records specific signatures associated with different malicious software, if it recognizes the pattern it is then flagged. While Anomaly-based defines characteristics that would normally be highly irregular to appear on a system (high execution times, system calls, high usage in the CPU) See Appendix A, figure 1.0 and 1.1 for the structure for Signature and Anomaly-based.

Intrusion Detection Systems are important in security management, as HIDS and NIDS allow administrators to monitor their system. Which is important as IDS software reveals details that administrators might not see.

The importance of IDS can be seen today as market research done by 'Markets and Markets' found that the market for IDS and IPS is projected to "increase up to 7.1 billion by 2024". The factors that have led to this was the increasing number of cyber-attacks, security threats and hacking attempts which had put pressure on the US government to increase the security of its system.[7] These factors are further backed up by a report done by Statista, which had found that the number of data breaches in 2015 had increased from 781 to 1473 data breaches in 2019.[8] This shows that IDS and IPS are needed in security management.

AIM

This report aims to show if Snort and OSSEC are reliable at detecting and alerting the administrator to any malicious software or activity happening on their system or over a network. The objectives are to launch different attacks and recon tools used in Kali Linux to see if these IDS programs can detect suspicious activity. As well, the project will explain how to set up Snort on Windows and OSSEC on Ubuntu.

3 PROCEDURE

OVERVIEW OF PROCEDURE

The experiments were conducted on Windows 7 and Ubuntu using a Virtual Machine (VMware). In the following sections will detail how to set up IDS on both Windows 7 and Ubuntu. The IDS programs that were used on these operating systems were Snort and OSSEC. The procedure is divided into two sections to provide a separate installation guide for installing each IDS program on two different operating systems. The procedures will detail on how to use the Intrusion Detection systems once they have been installed and configured properly. For virtual machine settings, see Appendix B

PROCEDURE (WINDOWS 7)

3.2.1 Software Requirements

There are a few programs that need to be installed. These programs will help in setting up Snort.

The applications that should be installed prior to installing Snort can be seen in Appendix C figure 3.0. For installing Snort, it is recommended to download a snapshot file. Which can be obtained by creating an account on Snorts website. This provides files that will be used later. Choose the current version of the snapshot for version 2.9. Then download the 2.9 version of Snort. Next, start the installation process by and click next when prompted to.

3.2.2 Installation

After Snort has been installed. Next, open the Snort rules snapshot by using 7zip and extracting the contents of rules and preproc_rules into the folders called rules and preproc_rules in the Snort directory. These folders may have to be created if they are not present.

The user should only extract the contents of these two folders, the ‘so_rules’ folder only offers pre-compiled versions of shared rules that are primarily used in Linux. To check that the installation was done correctly, open the command prompt and enter ‘cd \Snort\bin’. To check the version of Snort, enter ‘snort -V’. Next enter ‘snort -W’ to see what Network Adapters are available to use and take note of what number is assigned to which adapter.

3.2.3 Snort Configuration

Once Snort is properly installed, the next step is configuration. To start this, locate the Snort.conf file located in the etc folder in Snort’s directory. The configuration process is done in nine steps. For better visual representation each step of the configuration process will be in appendix C.

3.2.3.1 Set up the Network Variables

Step one is to set up the network variables that Snort will use to recognize our network and to declare the path routes for the rules in Snort's directory.

If Snort encounters an error, it will state which line in the configuration is causing the issue. Next, is to add the network address that is going to be monitored. To do this, enter the IP address with the subnets (/24) and below that line change the external network addresses from any to !$HOME_NET. This will stop other IP addresses from Snort scanning. See Appendix C figure 3.1.

Next, change the pathways for the Rules and preproc rules. Located on lines 104 & 106, the pathways should change from ../rules to c:\Snort\rules and the same for the preproc rules. A key detail to note is that the directories were a forward slash. Since Snort is used on Linux, their directories are forward slash. When changing the directories, it must be a backslash. Appendix C figure 3.1.

Next is to create a whitelist and blacklist. Snort does not come with a whitelist and blacklist by default, so it must be created. To do this, use notepad to create it. In notepad, it should contain two sentences that are commented stating that it's a whitelist to comment add a # at the beginning. Repeat this for the blacklist. To add an IP, just simply enter the IP addresses, make sure each IP address is per line.

When saving the file, save it as white.list and black.list in the rules folder. It is important to save it as all files, and not as a text file as it will not work. Once that is completed, next is to define the pathways for the white and blacklist as c:\Snort\rules for both. See Appendix C figure 3.2.

3.2.3.2 Configure the decoder

The decoder is set up by default, the line at the end of section 2 should be uncommented and altered to be c:\Snort\log. This will define the pathway for Snort to produce a log file. See Appendix C figure 3.3

3.2.3.3 Configure the base detection engine

There is no need for changing any settings in the base detection engine.

3.2.3.4 Configure dynamic loaded libraries

Next, edit the dynamicpreprocessor and dynamicengine directories. The changes to make are replacing the directory of the dynamic preprocessor with this 'c:\Snort\lib\snort_dynamicpreprocessor' and changing the directory of the dynamicengine to this 'c:\Snort\lib\snort_dynamicengine\sf_engine.dll'. Then comment on the dynamicdetection. See appendix C figure 3.4

3.2.3.5 Configure preprocessors

There are many preprocessors that Snort uses. Each preprocessor comes with a readme file, it's recommended to read those Readme files to set up the preprocessors. However this is not needed.

Next is to comment out all the rows in the Inline packet normalization preprocessor. This is because the preprocessor is used only when Snort is in IPS mode, but since Snort is being installed on Windows, this would normally cause an error as this is used in Linux. See appendix C figure 3.5

Depending on what version of Snort, the configuration file may contain the phrase ‘deflate Izma’. If it is in the configuration file, it’s normally found in the http_inspect preprocessor. Delete the ‘Izma’ as it refers to a data compression algorithm that is not usually installed on Windows.

3.2.3.6 Configure output plugins

Next, only uncomment the ‘include classification.config and include reference.config’ See appendix C figure 3.6.

3.2.3.7 Customize your rule set

This section is where any rules can be added in.

3.2.3.8 Customize preprocessor and decoder rule set

Next is to uncomment the following; include c:\Snort\preproc_rules\preprocessor.rules, include c:\Snort\preproc_rules\decoder.rules and include c:\Snort\preproc_rules\sensitive-data.rules

3.2.3.9 Customize shared object rule set

The ‘include threshold.conf’ should be uncommented.

3.2.4 Generating Alerts

This section demonstrates how to generate alerts and create alerts. These alert messages will appear in the console. If Snort detects these types of connections, the administrator will be alerted. The rules that were added are to detect TCP, UDP and ICMP connections. To do this, locate the local.rules file located in Snort’s rules folder. Open the file and enter the following lines.

```
22  alert icmp any any -> any any(msg:"ICMP Testing Rule"; sid:1000001; rev:1;)
23  alert tcp any any -> any 80 (msg:"TCP Testing Rule"; sid:1000002; rev:1;)
24  alert udp any any -> any any (msg:"UDP Testing Rule"; sid:1000003; rev:1;)
```

These rules simply look at what type of connection is occurring and checks which ports are being triggered.

Before starting Snort, the user needs to check what network adapters are on their system. To do this, change the directory in the command prompt to ‘cd \Snort\bin’, then enter ‘snort – W’. This should display the network adapters as shown.

```
Index   Physical Address        IP Address      Device Name     Description
-----   ----------------        ----------      -----------     -----------
    1   00:0C:29:4B:8D:15       0000:0000:fe80:0000:0000:0000:0102:3403 \Device\
NPF_{2DC1A96F-B58B-4094-A67D-353E8CEF1392}      Intel(R) PRO/1000 MT Network Con
nection
    2   00:00:00:00:00:00       0000:0000:fe80:0000:0000:0000:e059:dd49 \Device\
NPF_{890D2CBA-1E55-44A9-A62A-BB7CBDB56DC0}      MS LoopBack Driver
```

As it is shown, there are two network adapters, if unsure which network adapter to use, select one and if it’s incorrect then select the other one.

The user must now start up Snort. To do this, change the directory to this ‘cd \snort\bin’ then enter this command ‘snort -i 1 -c c:\Snort\etc\snort.conf -A console’. The number 1 is the network adapter that was chosen, if the command fails, replace the number with a different network adapter. The command should work if the user gets this message from the console.

```
Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 3.1  <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=2224)
```

To exit from Snort, enter control + c. This will then display the results. The results will detail what of alerts were triggered and other information will be displayed.

```
Packet I/O Totals:
   Received:           8
   Analyzed:           8 (100.000%)
    Dropped:           0 (   0.000%)
   Filtered:           0 (   0.000%)
Outstanding:           0 (   0.000%)
   Injected:           0
===============================================================================
Breakdown by protocol (includes rebuilt packets):
        Eth:           8 (100.000%)
       VLAN:           0 (   0.000%)
        IP4:           4 ( 50.000%)
       Frag:           0 (   0.000%)
       ICMP:           0 (   0.000%)
        UDP:           4 ( 50.000%)
        TCP:           0 (   0.000%)
        IP6:           0 (   0.000%)
    IP6 Ext:           0 (   0.000%)
   IP6 Opts:           0 (   0.000%)
      Frag6:           0 (   0.000%)
     ICMP6:            0 (   0.000%)
      UDP6:            0 (   0.000%)
      TCP6:            0 (   0.000%)
```

3.2.4.1 Optional step

For easier monitoring, it's recommended to use a syslog program. To do this, download a syslog server. During this procedure, the syslog server that was chosen was found on SourceForge. Next, download and execute the syslog file. The user will need to locate the following line of code in the Snort.conf file 'output alert_syslog: LOG_AUTH LOG_ALERT' and make the following changes as shown.

```
527   # syslog
528   output alert_syslog: host=127.0.0.1:514 LOG_AUTH LOG_ALERT
```

Next enter the command 'snort -i 1 -c c:\Snort\etc\snort.conf -s'. This will start up the syslog server. To note the syslog server should be running before Snort as this could lead to the syslog failing to startup.

3.2.4.2 The output of Snort on the console.

The console window as shown below details the alert messages created in the local rules file. To show the results of an ICMP alert, it can be done by pinging to a website. This leads to Snort monitoring the traffic being sent out from the machine.

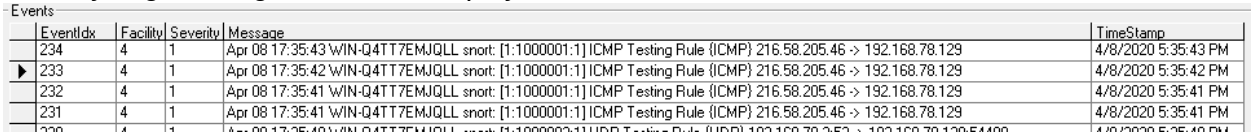
```
04/08-17:22:31.302869  [**] [1:1000001:1] ICMP Testing Rule [**] [Priority: 0]
ICMP> 216.58.204.36 -> 192.168.78.129
```

3.2.4.3 Analyzing the Console Output

By Analyzing the console output from the ping command, Snort can output the source and destination IPs of any messages. The rules that are created by the administrator in the local file in the rules folder help to identify the type of connection that is being sent out.

3.2.4.4 Output of Snort on the Syslog

If the syslog is being used, it will display a better look at the results.



Events

EventIdx	Facility	Severity	Message	TimeStamp
234	4	1	Apr 08 17:35:43 WIN-Q4TT7EMJQLL snort: [1:1000001:1] ICMP Testing Rule {ICMP} 216.58.205.46 -> 192.168.78.129	4/8/2020 5:35:43 PM
233	4	1	Apr 08 17:35:42 WIN-Q4TT7EMJQLL snort: [1:1000001:1] ICMP Testing Rule {ICMP} 216.58.205.46 -> 192.168.78.129	4/8/2020 5:35:42 PM
232	4	1	Apr 08 17:35:41 WIN-Q4TT7EMJQLL snort: [1:1000001:1] ICMP Testing Rule {ICMP} 216.58.205.46 -> 192.168.78.129	4/8/2020 5:35:41 PM
231	4	1	Apr 08 17:35:41 WIN-Q4TT7EMJQLL snort: [1:1000001:1] ICMP Testing Rule {ICMP} 216.58.205.46 -> 192.168.78.129	4/8/2020 5:35:41 PM

The syslog server allows an efficient way of monitoring traffic from Snort compared to the console. Which, the syslog server allows an administrator to look at the message and see what time and date the message was sent more clearly.

3.2.4.5 Rules

The rules seen in the local rules file will be used to generate alerts on the syslog.

```
alert icmp any any -> any any (msg:"ICMP Testing Rule"; sid:1000001; rev:1;)
alert tcp any any -> any 80 (msg:"TCP Testing Rule"; sid:1000002; rev:1;)
alert udp any any -> any any (msg:"UDP Testing Rule"; sid:1000003; rev:1;)
alert tcp any any -> 192.168.111.130 22 (msg: "NMAP TCP Scan";sid:10000005; rev:1;)
alert icmp any any -> 192.168.111.130 any (msg: "NMAP ping sweep Scan"; dsize:0;sid:10000004; rev: 1;)
alert tcp any any -> 192.168.111.130 22 (msg:"Nmap XMAS Tree Scan"; flags:FPU; sid:1000006; rev:1;)
alert tcp any any -> 192.168.111.130 22 (msg:"Nmap FIN Scan"; flags:F; sid:1000008; rev:1;)
alert udp any any -> 192.168.111.130 any ( msg:"Nmap UDP Scan"; sid:1000010; rev:1;)
```

The rules are based on using Nmap commands on a Kali Linux. The purpose of these rules is to show how Snort will detect commands, such as XMAS Tree Scan, UDP Scan and FIN Scan. In the results section will show what commands to enter and the results.

PROCEDURE (UBUNTU)

3.3.1 Required packages

Before installing OSSEC, some packages need to be installed without them prevents OSSEC from running. First, is to check that the version of Ubuntu is updated. Next is to install different packages. See appendix C figure 3.5 for the list packages to install. Once these packages have been installed, enter ‘sudo apt-get upgrade -y’. This command will check if the packages need an update.

Next is to install a web server. To do this enter ‘sudo apt-get install apache2’ and then enter y. To verify the installation, enter ‘https://’ then your IP address. In this procedure, Apache is being used as a web manager. Then enter ‘sudo systemctl restart apache2’ to restart the server.

3.3.2 Installing OSSEC

Next is to install OSSEC, to do this enter the following command ‘wget https://github.com/ossec/ossec-hids/archive/3.3.0.tar.gz’ This command will retrieve the repository of OSSEC from GitHub. Then extract the contents in the tar file. To do this enter the command ‘tar -xvzf 3.3.0.tar.gz’.

In the terminal, change the directory to ‘cd ossec-hids-3.3.0’. Finally, to begin the process of installing OSSEC enter the command ‘sh install.sh’. Then OSSEC will require the administrator to answer questions on how they want to set up OSSEC on their machine.

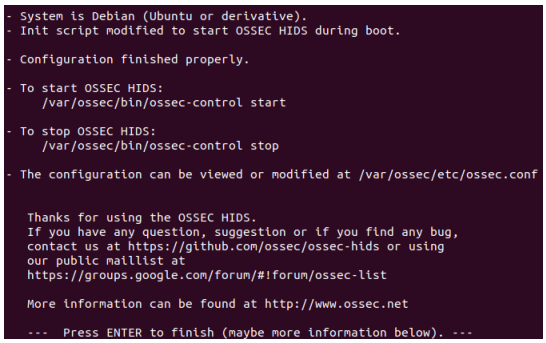
The questions that are asked are what language and what kind of installation. For this procedure, it was local that was selected for the installation.

Next was to choose where in the system OSSEC should store its files, the default location is the OSSEC folder, to confirm this directory just press enter.

The next question asks if the administrator would like to opt-in to an email notification. This is optional but for now, enter n. The next questions ask for permission to run an integrity check daemon and root check, press enter to enable these features. This is because the integrity check allows the integrity of the system. This is done by checking the MD5/SHA1 checksum. Allowing root check provides endless monitoring and alerting the administrator.

Next, OSSEC asks if the administrator wants to enable an active response, which permits the user to run a command based on events that have occurred. Which is an effective protection method to stop any suspicious IP address or to restrict permissions on users who shouldn't have it. The active response is a powerful tool to enable, as it allows an administrator to manage how permissions are distributed and to maintain the security of their system.

Next, OSSEC asks if the administrator would like to enable the firewall-drop response. Enabling this will then ask the user if they would like to whitelist more IPs on the OSSEC system. The default IP is the local hosts. After the setup questions have been answered, the terminal then displays how to start and stop OSSEC and that configuration can be viewed/modified in the OSSEC.conf. The directory which is shown here.

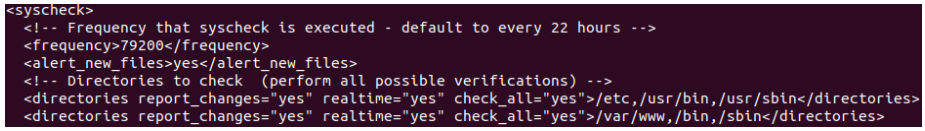


To test OSSEC, enter the start command as shown above. This will then display a series of messages saying 'Started ossec-maild, started ossec-monitored' with other messages alongside it.

3.3.3 Configuring OSSEC

With OSSEC installed the next is to configure the files. To begin the process, enter the command 'sudo nano /var/ossec/etc/ossec.conf'. This will open a text editor and, in the file, the first change to make is allowing OSSEC to alert the server that a new file has been added in. To do this locate the line '<frequency> 79200</frequency>', below that line add '<alert_new_files>yes</alert_new_files>'.

Next is to allow OSSEC to send real-time alerts to the user, by default it does not do this. To change this, change the directories and add in the 'report_changes', 'realtime' and 'check_all' as shown below.



Then save and exit the file.

Next is to modify the rules file in the OSSEC folder. To do this, enter the command ‘sudo nano /var/ossec/rules/local_rules.xml’ This will open the text editor again and then the user should add the following rule 554 under rule 100001.

```
<rule id="100001" level="0">
  <if_sid>5711</if_sid>
  <srcip>192.0.2.1</srcip>
  <description>Example of rule that will ignore sshd </description>
  <description>failed logins from IP 1.1.1.1.</description>
</rule>

<rule id="554" level="7" overwrite="yes">
  <category>ossec</category>
  <decoded_as>syscheck_new_entry</decoded_as>
  <description>File added to the system.</description>
  <group>syscheck,</group>
</rule>
```

Then save and close that file, to make sure the changes are applied, restart OSSEC. To do this enter the command ‘/var/ossec/bin/ossec-control restart’.

3.3.4 Installing OSSEC Web UI

This section is installing the Web user interface for OSSEC, which is an essential and powerful tool to monitor the traffic.

The first step is to retrieve the repository from GitHub. To do this enter this ‘wget https://github.com/ossec/ossec-wui/archive/master.zip‘. Next is to extract the contents, which can be done by entering ‘unzip master.zip’.

The next step is to move the extracted directory to the Apache web directory, which was made at the beginning. To do this enter ‘mv ossec-wui-master /var/www/html/ossec’. Once that is done change the directory to the apache web directory by entering ‘cd /var/www/html/ossec’. To start the installation, enter ‘./setup.sh’.

Now, the user should be prompted to create a username and password, then it’ll ask the user to restart the webserver. To do this simply enter ‘systemctl restart apache2’

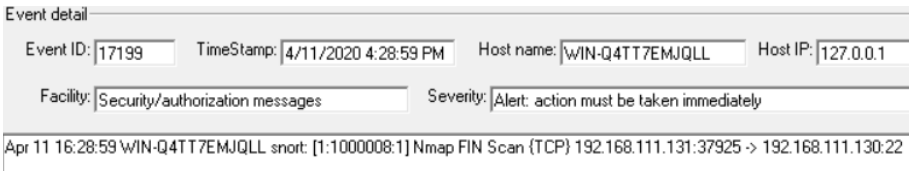
Once the user has completed that step, that is OSSEC fully installed.

4 RESULTS

RESULTS FOR SNORT

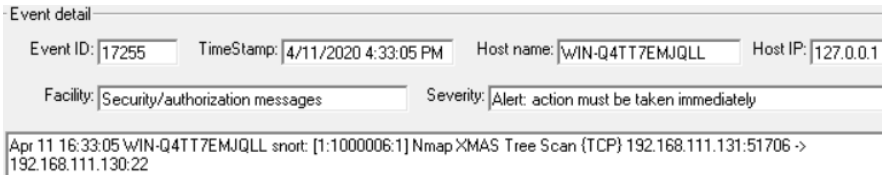
4.1.1 NMAP FIN Command and Results

```
Nmap done: 1 IP address (1 host up) scanned in 0.41 seconds
root@kali:/home/kali# nmap -sF -Pn  -p22 192.168.111.130
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-11 11:30 EDT
Nmap scan report for 192.168.111.130
Host is up (0.00040s latency).

PORT   STATE         SERVICE
22/tcp open|filtered ssh
MAC Address: 00:0C:29:BC:69:17 (VMware)
```



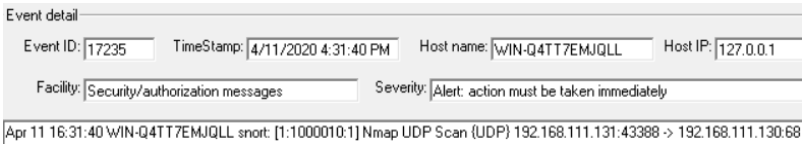
4.1.2 NMAP XMAS Tree Command and Results

```
Nmap done: 1 IP address (1 host up) scanned in 0.79 seconds
root@kali:/home/kali# nmap -sX -Pn -p22 192.168.111.130
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-11 09:25 EDT
Nmap scan report for 192.168.111.130
Host is up (0.0012s latency).

PORT   STATE         SERVICE
22/tcp open|filtered ssh
MAC Address: 00:0C:29:BC:69:17 (VMware)
```



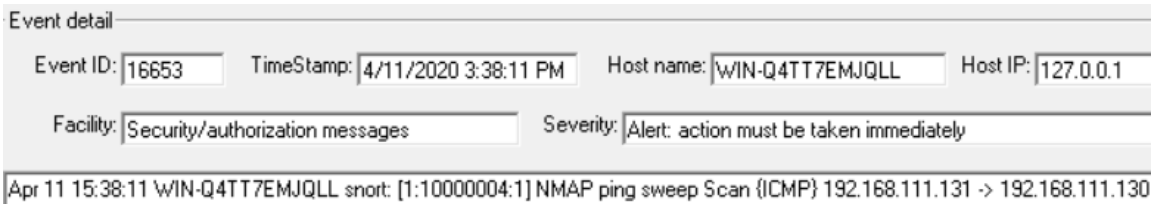
4.1.3 NMAP UDP Scan Command and Results

```
root@kali:/home/kali# nmap -sU -p68 192.168.111.130
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-11 11:31 EDT
Nmap scan report for 192.168.111.130
Host is up (0.00040s latency).

PORT   STATE         SERVICE
68/udp open|filtered dhcpc
MAC Address: 00:0C:29:BC:69:17 (VMware)
```



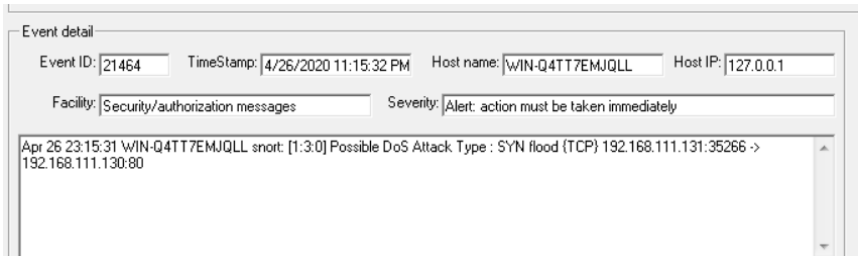
4.1.4 NMAP Ping Sweep Command and Results

```
root@kali:/home/kali# nmap -sP 192.168.111.130 --disable-arp-ping -Pn
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-11 12:30 EDT
Nmap scan report for 192.168.111.130
Host is up.
Nmap done: 1 IP address (1 host up) scanned in 0.02 seconds
```



4.1.5 DDOS Test: using hping3 to target Windows machine IP at port 80

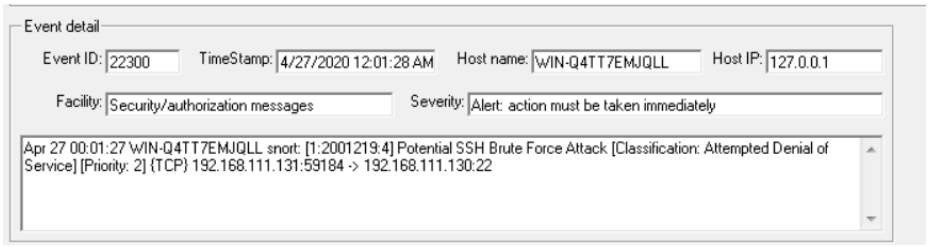
```
alert tcp any any -> $HOME_NET 80 (flags: S; msg:"Possible DoS Attack Type : SYN flood"; flow:stateless; sid:3; detection_filter:track by_dst, count 20,
seconds 10;)
```



4.1.6 SSH Brute force rule triggered from an Nmap scan

```
kali@kali:~$ nmap -v -Pn 192.168.111.130
```

```
alert tcp any any -> $HOME_NET 22 (msg:"Potential SSH Brute Force Attack"; flow:to_server; flags:S; threshold:type threshold, track by_src, count 3, seconds
60; classtype:attempted-dos; sid:2001219; rev:4; resp:rst_all;)
```



OSSEC RESULTS

4.2.1 OSSEC Detection on creation of files

File created called index.html

```
root@ubuntu:/home/toor# touch /home/toor/Desktop/index.html
root@ubuntu:/home/toor# nano /home/toor/Desktop/index.html
root@ubuntu:/home/toor# rm /home/toor/Desktop/index.html
```

4.2.2 OSSEC Detecting that a new file has been added into the system

Level: 7 - File added to the system. **2020 Apr 24 13:03:57**
Rule Id: 554
Location: ubuntu->syscheck

New file '/var/www/html/index.html' added to the file system.

4.2.3 OSSEC Detecting that the client has switched to root access

Level: 3 - Successful sudo to ROOT executed **2020 Apr 24 13:05:24**
Rule Id: 5402
Location: ubuntu->/var/log/auth.log
User: toor

Apr 24 13:05:23 ubuntu sudo: toor : TTY=pts/1 ; PWD=/home/toor ; USER=root ; COMMAND=/bin/su

4.2.4 OSSEC detecting that a file has been alerted (Integrity file check)

Level: 7 - Integrity checksum changed. **2020 Apr 25 06:03:23**
Rule Id: 550
Location: ubuntu->syscheck

Integrity checksum changed for: '/bin/ps'
Size changed from '133432' to '62920'
Ownership was '0', now it is '122'
Group ownership was '0', now it is '114'
Old md5sum was: '558edc26f8a38fa9788220b9af8a73e7'
New md5sum is : 'ced323b51dc984f66c2695d8fd6a2368'
Old sha1sum was: '3024d44e580e9c67f32f6c585d50e2a6cc9a7cac'
New sha1sum is : '46efcecf8383aee782f62bfc599edaa2e3c29903'
What changed:
Binary files /var/ossec/tmp/syscheck-changes-ced323b51dc984f66c2695d8fd6a2368-1587819803 and /var/ossec/tmp/syscheck-changes-558edc26f8a38fa9788220b9af8a73e7-1587756952 differ

4.2.5 OSSEC Detecting rootkit

```
root@ubuntu:/home/toor# git clone https://github.com/CCrashBandicot/shv5.git
Cloning into 'shv5'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 11 (delta 2), reused 11 (delta 2), pack-reused 0
Unpacking objects: 100% (11/11), done.
root@ubuntu:/home/toor# cd shv5/
root@ubuntu:/home/toor/shv5# chmod 777 setup
root@ubuntu:/home/toor/shv5# ./setup
```

```
------------------------------------------------------------------------
[sh]# Just ignore all errors if any !
[sh]# ============================== Backdooring completed in :5 seconds
./setup: line 813: /sbin/syslogd: No such file or directory
root@ubuntu:/home/toor/shv5#
```

```
root@ubuntu:/home/toor/shv5# cd /var/ossec
root@ubuntu:/var/ossec# ./bin/agent_control -lc

OSSEC HIDS agent_control. List of available agents:
   ID: 000, Name: ubuntu (server), IP: 127.0.0.1, Active/Local

root@ubuntu:/var/ossec# ./bin/agent_control -r -u 001
2020/04/26 11:12:13 agent_control(1751): ERROR: File client.keys not found or empty.
root@ubuntu:/var/ossec# ./bin/agent_control -r -u 000

OSSEC HIDS agent_control: Restarting Syscheck/Rootcheck locally.
root@ubuntu:/var/ossec#
```

Level: 7 - Host-based anomaly detection event (rootcheck). **2020 Apr 25 10:14:13**
Rule Id: 510
Location: ubuntu->rootcheck

Rootkit 'shv5' detected by the presence of file '/usr/lib/libsh'.

Level: 7 - Host-based anomaly detection event (rootcheck). **2020 Apr 25 10:14:13**
Rule Id: 510
Location: ubuntu->rootcheck

Rootkit 'shv5' detected by the presence of file '/lib/libsh.so'.

5 DISCUSSION

GENERAL DISCUSSION

5.1.1 Discussion Snort

The first results detail the use of various Nmap commands used on Kali Linux targeting the Windows machine that Snort was running on. The goal of this procedure was to see if Snort could detect Nmap which by the results show that it can. The following Nmap commands were detected by triggering the rules that were implemented. Snort was able to detect Nmap FIN, XMAS, UDP and Ping sweep scans and TCP scans. The results of the syslog detail the attacker's IP address, which in this case was 192.168.111.130. The user could use this information to add the attacking IP address to Snort's blacklist feature. Doing this can prevent the attacker from further scanning the machine. The syslog also details what time the attack took place, which a user can use to track the duration of the attack and allows for real-time monitoring.

The next result was to see If Snort can detect a DOS attack with the right rules in place. As shown in the result sections, Snort can detect a DOS attack. However, during the procedure, the syslog tended to crash while the attack was still going and the way to fix that was by restarting the virtual machine. The only rule implemented was a simple rule that would display a message when the Windows machine was receiving TCP packets with no data on port 80. The process of attacking the machine was done by using the hping application in Kali. The command that was entered would flood a series of TCP packets with an SYN flag to the Windows machine. The command would also target the machine at port 80, which Snort would detect. Although the user could alter the rule to detect hping3 scans at any port of their choosing.

The one issue that occurred was that the syslog program would display the alerts but then crash. This is believed to be caused by the application not able to handle the number of alerts occurring at once. As the hping3 command led to the dos alert being triggered constantly, which could explain why the application crashed. An alternative could be to use the console to display the alerts or use a much better syslog server with better support.

Overall, configuring Snort to the right specification can detect various intrusions over a network. Snort was able to detect NMAP commands sent by Kali and the results displayed the attacker's IP address along with which ports the attacker was using. This has met the aim that Intrusion Detection systems like Snort on Windows can detect different intrusions.

5.1.2 Discussion OSSEC

The results for OSSEC help to demonstrate the features that are offered to alert an administrator if the integrity or protection of their system has been compromised or any files that have been added to the system.

The first procedure was to trigger an alert to the user that a file has been added into the system. To do this a file was created called index.html and after some attempts, OSSEC was able to detect that a file called 'Index.html' was added to the system. With this information, a user could track down the suspicious file by entering the file name.

Another feature that OSSEC provides is alerting an administrator if user permissions have changed. For example, in the procedure, it shows that a user has switched to root privileges and OSSEC does also display if a user has successfully logged into an account. This information is important as It could alert the administrator that an account may have been compromised or an account has permissions that they should not have.

The next procedure demonstrates the Integrity files feature. Which alerts the user that a file in the system has been altered. OSSEC checks this by comparing the details of the file before and after it was alerted. In the procedure, it is shown that OSSEC checks the sha1mum values, the size of the file and any permission changes. But importantly, OSSEC shows which file specificity has been alerted, which in the procedure above shows that a binary file had been alerted.

The last procedure (4.25) was done to demonstrate that OSSEC could detect a rootkit in the system files. To do this, a Shv5 rootkit that was coded in 2003 was used to test this. The rootkit was granted all the permissions on the system to see how well OSSEC could detect it. With the rootkit installation process done, the next step was to reset syscheck. Doing this would result in OSSEC checking if there was any rootkit software installed. The following commands after rootkit were installed were done to find out the ID of the OSSEC server. This was done to restart the syscheck. After the restart OSSEC was able to detect and name that rootkit shv5 was detected on the Ubuntu server. Which listed the rootkit as a Host Anomaly detection, which is an example of a HIDS.

Overall, OSSEC was able to detect changes made to the Ubuntu system. OSSEC was able to alert the administrator that files were added to the system and displayed the directory of where the files are contained. The IDS software was able to detect an intrusion with the example of the rootkit and enabled monitoring of any files being added into the system.

COUNTERMEASURES

5.2.1 Countermeasures/future work for Snort

The possible countermeasures that an administrator could undertake is starting up Snort in Intrusion Prevention System mode (IPS). Which allows for drop and reject rules to be triggered when Snort is running. The purpose of these rules is to drop the packets being sent to the Windows machine and alerting the administrator.

The rules for drop and reject can be the same as the alert rules. To do this at the beginning of creating a rule, inserting drop or reject at the beginning will let Snort recognize that these are drop rules. The other possible solutions could be to upgrade security measures, such as firewalls and anti-virus protection software as well. To further strengthen the security of the machine. Further work that could be done is further going into detail on the configurations that Snort offers.

5.2.2 Countermeasures/future work for OSSEC

The possible countermeasures for OSSEC could be to use Security information and event management (SIEM) software. Such as AlienVault, which uses the OSSEC agents for Host Intrusion Detection. The purpose of this is for OSSEC agents to collect information, such as root check, log monitoring and report the information back to the AlienVault server. AlienVault allows for better configuration and monitoring of multiple OSSEC agents.

CONCLUSIONS

In conclusion, it is found that with the right configuration done by the administrator, the IDS software that was tested – being Snort and OSSEC – it is determined that Intrusion Detection Software is reliable in monitoring a user's system and providing the necessary information and features for an administrator to take action in protecting their system from attacks or to prevent any suspicious activity from ever happening again.

FUTURE WORK

5.4.1 Future Work for Snort

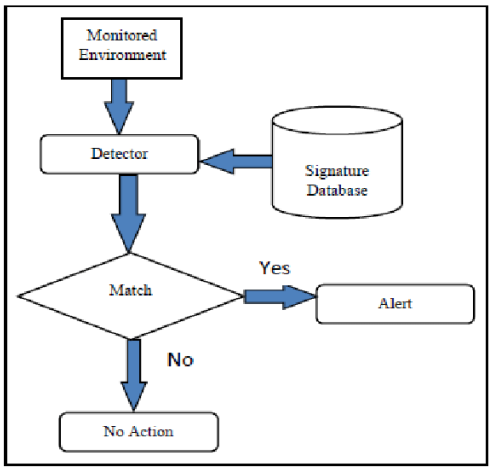
With more time and resources, would allow a chance to look further into the configuration of Snort on a network level. By that setting up Snort on multiple clients and connecting the clients to a SIEM service rather than using a syslog application or console to output the results. This would present a better opportunity to monitor multiple clients running Snort and seeing how the clients will react when a suspicious activity has been detected.

5.4.2 Future Work for OSSEC

Future work that could be conducted is running OSSEC on multiple machines at once and connecting the machines to a SIEM application, such as AlienVault rather than using OSSEC's SIEM service.

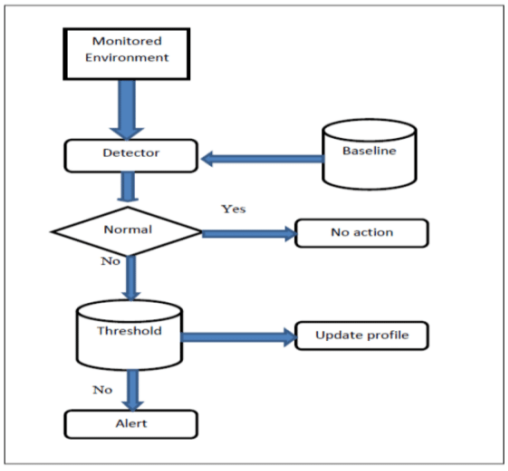
APPENDIX A

Signature based Figure 1.0



(Mudzingwa, D. and Agrawal, R, 2012, Section III, figure 3)

Anomaly-based Figure 1.1

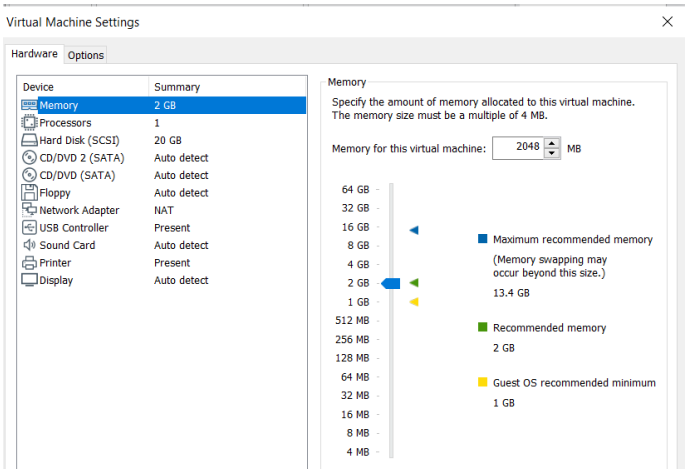


Source for figure 1.0 and 1.1: Mudzingwa, D. and Agrawal, R., 2012, March. A study of methodologies used in intrusion detection and prevention systems (IDPS). In *2012 Proceedings of IEEE Southeastcon* (pp. 1-6). IEEE.

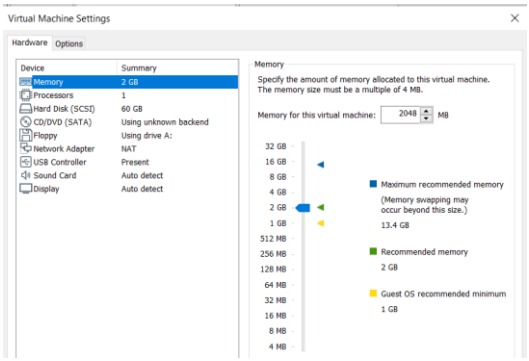
(Mudzingwa, D. and Agrawal, R, 2012, Section III, figure 2)

APPENDIX B

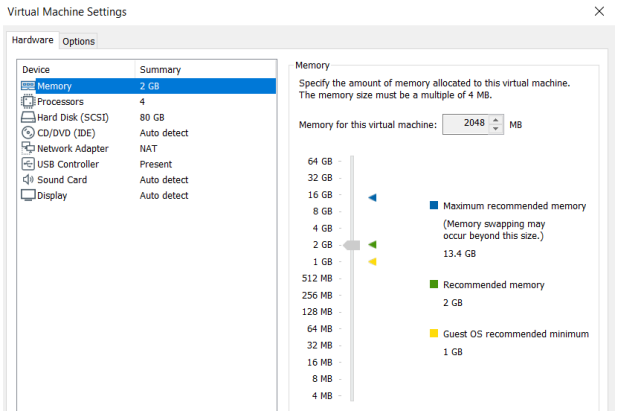
Windows 7 Settings Figure 2.0



Ubuntu Settings Figure 2.1



Kali Linux Settings Figure 2.2



APPENDIX C

Figure 3.0

```
7-Zip - used for extracting the contents in the Snort SnapShot
Sublime Text 3 - used for editing the Snort.conf and local file
WinPcap - used with Snort
```

Figure 3.1

```
44  # Setup the network addresses you are protecting
45  ipvar HOME_NET 192.168.78.137/24
46
47  # Set up the external network addresses. Leave as "any" in most
    situations
48  ipvar EXTERNAL_NET !$HOME_NET
```

Figure 3.2

```
104  var RULE_PATH c:\Snort\rules
105  #var SO_RULE_PATH ../so_rules
106  var PREPROC_RULE_PATH c:\Snort\preproc_rules
107
108  # If you are using reputation preprocessor s
109  # Currently there is a bug with relative pat
110  # not relative to snort.conf like the above
111  # This is completely inconsistent with how o
112  # Set the absolute path appropriately
113  var WHITE_LIST_PATH c:\Snort\rules
114  var BLACK_LIST_PATH c:\Snort\rules
```

Figure 3.3

```
184  # Configure default log directory for
     line options (-l)
185  #
186   config logdir: c:\Snort\log
```

Figure 3.4

```
246  # path to dynamic preprocessor libraries
247  dynamicpreprocessor directory c:\Snort\lib\snort_dynamicpreprocessor
248
249  # path to base preprocessor engine
250  dynamicengine c:\Snort\lib\snort_dynamicengine\sf_engine.dll
251
252  # path to dynamic rules libraries
253  #dynamicdetection directory /usr/local/lib/snort_dynamicrules
```

Figure 3.5

```
263  # Inline packet normalization. For more info
264  # Does nothing in IDS mode
265  #preprocessor normalize_ip4
266  #preprocessor normalize_tcp: ips ecn stream
267  #preprocessor normalize_icmp4
268  #preprocessor normalize_ip6
269  #preprocessor normalize_icmp6
```

Figure 3.6

```
534  include classification.config
535  include reference.config
```

Figure 3.7

```
sudo apt-get install build-essential
sudo apt-get install -y libssl-dev
sudo apt-get install libevent-dev
sudo apt-get install libz-dev
sudo apt-get install -y libpcre2-dev
sudo apt install php libapache2-mod-php
```

6 REFERENCES & BIBLIOGRAPHY

REFERENCES

[1] Snort - https://www.snort.org/ (Accessed: 3st March)

[2] OSSEC - https://www.ossec.net/ (Accessed 8th March)

[3] Imperva – Intrusion detection and intrusion prevention https://www.imperva.com/learn/application-security/intrusion-detection-prevention/ (Accessed: 1st March 2020)

[4] Heenan, R. and Moradpoor, N., 2016, May. A survey of Intrusion Detection System technologies. In *PGCS 2016: the first post graduate cyber security symposium. The Cyber Academy, Edinburgh Napier University. 10th May*. Edinburgh Napier University. (pp. 1-5) (Last Accessed: 3rd March 2020)

[5] Rapid7, 2017, Jan 11 The Pros & Cons of Intrusion Detection Systems Rapid Blog https://blog.rapid7.com/2017/01/11/the-pros-cons-of-intrusion-detection-systems/

(Last Accessed: 30th March)

[6] Kovanen, T., David, G. and Hämäläinen, T., 2016. Survey: Intrusion detection systems in encrypted traffic. In *Internet of Things, Smart Spaces, and Next Generation Networks and Systems* (pp. 281-293). Springer, Cham.

(Last Accessed 4th March 2020)

[Appendix A] Mudzingwa, D. and Agrawal, R., 2012, March. A study of methodologies used in intrusion detection and prevention systems (IDPS). In *2012 Proceedings of IEEE Southeastcon* (pp. 1-6). IEEE.

(Last Accessed 4th March)

[7] Market and Markets, December 2019 - Intrusion Detection and Prevention Systems Market by Component

(Last Accessed 5th April)

https://www.marketsandmarkets.com/Market-Reports/intrusion-detection-prevention-system-market-199381457.html

(Last Accessed 5th March 2020)

[8] Statista – Annual Number of data breaches and exposed records in the United States from 2005 to 2019. Published in 2020 https://www.statista.com/statistics/273550/data-breaches-recorded-in-the-united-states-by-number-of-breaches-and-records-exposed/

(Last Accessed 13th March 2020)

BIBLIOGRAPHY

Heenan, R. and Moradpoor, N., 2016, May. A survey of Intrusion Detection System technologies. In *PGCS 2016: the first post graduate cyber security symposium. The Cyber Academy, Edinburgh Napier University. 10th May*. Edinburgh Napier University.
(Last Accessed: 3rd March 2020)

Kovanen, T., David, G. and Hämäläinen, T., 2016. Survey: Intrusion detection systems in encrypted traffic. In *Internet of Things, Smart Spaces, and Next Generation Networks and Systems* (pp. 281-293). Springer, Cham.

(Last Accessed 4th March 2020)

Mudzingwa, D. and Agrawal, R., 2012, March. A study of methodologies used in intrusion detection and prevention systems (IDPS). In *2012 Proceedings of IEEE Southeastcon* (pp. 1-6). IEEE.
(Last Accessed 4th March)

Rapid. (2017, Jan 11) The Pros & Cons of Intrusion Detection Systems retrieved on 3rd March 2020

(Last Accessed: 30th March)

https://blog.rapid7.com/2017/01/11/the-pros-cons-of-intrusion-detection-systems/

Statista (January 2020) – Annual Number of data breaches and exposed records in the United States from 2005 to 2019.

(Last Accessed 13th March 2020)

https://www.statista.com/statistics/273550/data-breaches-recorded-in-the-united-states-by-number-of-breaches-and-records-exposed/

Market and Markets (December 2019) - Intrusion Detection and Prevention Systems Market by Component

(Last Accessed 5th April). https://www.marketsandmarkets.com/Market-Reports/intrusion-detection-prevention-system-market-199381457.html

OSSEC Documentation https://www.ossec.net/docs/manual/installation/index.html

(Last Accessed: 5th April)

Snort Documentation - https://www.snort.org/documents

(Last Accessed: 7th April)

Bray, R., Cid, D. and Hay, A., 2008. *OSSEC host-based intrusion detection guide*. Syngress. (Last Accessed: 15th April)

Caswell, B. and Beale, J., 2004. *Snort 2.1 intrusion detection*. Elsevier. (Last Accessed: 5th April)

Security Architecture -Intrusion Detection Systems: Learning with Snort https://www.securityarchitecture.com/learning/intrusion-detection-systems-learning-with-snort/getting-and-installing-tools/

(Last Access March 8th)