**Snort Intrusion Detection and Prevention System**

Sandeep Chinthapatla

# Abstract

Snort generally is a most commonly used Intrusion detection and prevention system. In this report, we discuss all the minute elements that are required from installing snort to configuring it as a full-fledged Intrusion detection and prevention system. We also give an overview of functioning of Snort by performing several attacks such as cross-site scripting and SQL injection attack. By whole we present a complete step by step process of deploying a complete package of Snort along with configuration of both the attacker and the victim.

*Keywords:* Snort, Barnyard2, Base, Bodgeit store

# 

# Intrusion Detection and Prevention System:

Intrusion detection can be defined as the process for monitoring all the events that occur in a computer system or network. Based on these events we analyze them for signs of any threats and violations of policies.

We can generally define Intrusion detection system (IDS) as a software package that is responsible for detection of threats across the network or system while Intrusion prevention system (IPS) as a software responsible for stopping all the events. (NIST, n.d.)

Nowadays we have systems which run both as an Intrusion detection and prevention systems. Among that we can define some of the open-source software’s as Snort, Suricata, Bro etc.

Intrusion detection systems can be generally classified into: (Wikipedia, n.d.)

1. Host-based Intrusion detection system
2. Network Intrusion detection system

## Host-based Intrusion detection system:

Host-based as the name suggests work on host devices. It monitors all the incoming and outgoing packets and alerts if any suspicious activity is observed. As it works on the host system it has a previous record of all the files. Whenever someone tries to make changes to the files it creates a alert notifying that the files have been changed.

## Network Intrusion detection system:

Network Intrusion detection system as the name suggests we concentrate on one point which we can place at any part of the network and monitor. If we find any changes in the behavior, then an alert will be generated.

## Detection methods:

For now, we have two methods of detection as follows:

#### Signature-based:

In this the detection of attacks is done by looking at patterns such as byte sequences in network traffic or known malicious instruction sequences used by malware. This can be easily linked to anti-virus software which we use in our systems. The way it works is simple. It has pre-defined patterns stored in its database. So, if any changes in behavior is raised then it detects as malicious. If any new attacks are done it can’t detect because it does not have the new pattern stored in its database. That is one of the drawbacks that we can consider for Signature-based Intrusion detection systems.

#### Anomaly-based:

Due to the increase in newer form of attacks and malware this type of detection was introduced. This is primarily introduced to detect newer attacks which are unlike the signature-based. The only problem with this type of detection approach is it might suffer from false positives.

Now from the above concepts we clearly understand what an Intrusion Detection and Prevention System usually is. Below we will be giving a precise information ranging from installation of machines such as victim, attacker and “Snort” to deploying it and performing attacks to see the working of Snort.

# Inventory required:

The Inventory which we require for the Snort Intrusion Detection and Prevention System is given in below:

## (a) Hardware Requirements:

**1. Host machine requirements**:

1. RAM: 8/16 GB
2. Processor: Intel Core i5/i7
3. HDD: 1TB (minimum)
4. OS: WINDOWS 8/10

## (b) Software Requirements:

VMware Workstation: For installing various OS to run the project.

**Operating System needs:**

1.KALI 2017.2 (Virtual Machine): We are going to use this as our attacking machine.

RAM: 2GB of RAM. (by default)

HDD: 20GB (by default)

2. Ubuntu (Virtual Machine): We are using two Ubuntu machines where one will be the victim and the other would be running SNORT.

RAM: 2GB of RAM (by default)

HDD: 20GB (by default)

## Virtual Machine:

It is an operating system which is installed in a software machine and acts as a physical machine. The user will have same experience as they are working on a real hardware machine. One of the advantages with virtual machine is that we can install and operate multiple operating systems simultaneously in a single virtual machine. Additionally, operating systems can be tested ideally. (Tech target, n.d.)

#### Ubuntu:

It is one of the Linux operating system which is an open source software. Not only for the professional use it is also available for community support. It is encouraged for public to use, improve it and spread an open source tool, as it is entirely based on the principles of open source software. It is suitable for server as well desktop use. (Ubuntu, n.d.)

#### Kali:

It is a specially designed open source software for Digital Forensics and Penetration testing. It is Debian-derived distribution system. (Wikipedia, n.d.)

## Snort:

It is an open source intrusion detection system. It tracks, analyses and logs the packets in the live traffic which flows through the network. (Snort-org, n.d.)Apart from analysis it does searching and matching. Various attacks such as buffer overflows, stealth port scanning, and CGI attacks are detected by Snort. Snort can also be utilized as a packet sniffer such as TCP dump, a packet logger, logging device for a network. On the other hand, snort also acts as an Intrusion Prevention System in inline mode. (Talos intelligence, n.d.)

Snort can be configured in three modes. They are: (Wikipedia, n.d.)

1. Sniffer mode: In this mode it will read all the packets from the network and displays it on the console.

2. Packet logger mode: This mode is used to write all the logs to the disk. In this if we specify a network it will capture all the incoming packets and stores them into the directories specified. (Snort manual, n.d.)

3. Intrusion detection mode: In this project we use this mode. In this it will monitor all the network traffic and analyze it based on the rules defined by the user. After this it will perform any action if provided.

#### Components of Snort:

Snort has five components as: (exciting ip, n.d.)

##### **Packet decoder:**

It is the first component which collects packets from different network interfaces and prepares the packets to be preprocessed.

##### **Pre-processor:**

Its primary use is to arrange the packets and modify them before it is sent to the next stage of analysis done by the detection engine. They can detect various anomalies ranging from packet defragmentation to HTTP URL decoding. Apart from these it can detect several other activities too.

##### **Detection engine:**

It is the heart of Snort. It’s mainly responsibility is to analyze all the packets passing through snort for checking any intrusion using the rules defined. In our case we define our own rules and this detects it for any intrusion occurring. It can dissect a packet and apply the rules individually to the least component too.

##### **Logging and Alerting System**:

Based on the above phase that is detection engine it either logs the activity or generates an alert.

***Output modules or plug-ins****:*

It is used to control the type of output produced by the logging and alerting system. Its functions include generation of log reports, generating alerts etc.

#### JAVA:

Java is a computer software developed by Sun microsystems. It is used for developing application software and deploying in a cross-platform computing environment. (wikipedia, n.d.)

###### **JDK and JRE:**

JDK can be referred as Java development kit which contains JRE which is Java runtime environment.

###### **Why Java is required?**

In the installation of Bodgeit store in the victim machine we need JDK and JRE so that we can run Apache Tomcat to host. We come back to Bodgeit store in the later part of the report where we give an intro to Bodgeit store and describe the setup of victim machine.

#### MYSQL:

MYSQL is a popular open source database. It is very easy to use, scalable and has high performance which can meet the standards of the users. (Wikipedia, n.d.)

###### **Why MYSQL is required?**

Snort can directly log into MySQL in real-time to access database. Barnyard2 utility tool collects all the unified2 log files from Snort and pushes it into the database then projects to BASE for visualization. So, MYSQL is a critical component.

#### Apache Tomcat:

Apache Tomcat software is an open source implementation of the Java Servlet, Java Server Pages, Java Expression Language and Java Web Socket technologies. The Java Servlet, Java Server Pages, Java Expression Language and Java Web Socket specifications are developed under the Java Community process. (Apache tomcat, n.d.)

###### **Why Apache Tomcat is required?**

For hosting the Bodgeit store in the victim machine and running BASE in the Snort machine we require Apache Tomcat.

#### Barnyard2 and why it is required?

It is one of the open source interpreter for the binary output files of Snort unified2. It is basically a Snort’s output system. A special binary format output which is generated by snort is **unified2**. Then Barnyard reads the file which is sent by Snort and resends the file to the database. The main purpose of Barnyard is, in efficient manner it allows snort to write to disk and it will not parse the binary data into various formats, by which snort will not miss the network traffic.

**PHP and why it is required?**

PHP is a server-side scripting language. It’s main use if for web development and creation of dynamic web pages. The programming language required for BASE is PHP, so we use it. (wikipedia, n.d.)

#### BASE and why it is required?

BASE is used for visualization of the log files. It collects all the log files from Barnyard2 and displays it in a rich graphic content form. The graphic content is nothing but charts, bar diagrams etc. By this we can easily view and analyze the log files.

#### Bodgeit store:

The BodgeIt Store created by Psiinon is a vulnerable web application. It consists of a wide variety of vulnerabilities and is NOT intended to be hosted on a production environment. (Infosec, n.d.)

**Why Bodgeit store?**

We use Bodgeit store in the victim machine so that we can perform attacks from attacking machine to the victim.

#### ELK stack:

ELK stack can be referred as Elasticsearch, Logstash and kibana. Each has its own functionality. They are open software tools offered by the company called elastic. Elasticsearch is built in search, analytics engines and information retrieval library for enabling data retrieval and storage. (Generating artificial snort alerts and implementing SELK:The Snort-elastic-logstash-kibana, 2017)

###### **Elasticsearch:**

As the name suggest elasticsearch is used to retrieve the logs from logstash.

###### **Logstash:**

Logstash provides data collection and processes the data in order to manage the event and logs.

###### **Kibana:**

Kibana reads the elastics search data, and offers graphical interface to visualize data in the elasticsearch.. This will generate, parse, store and visualize and analyze any snort alert with the Elasticsearch-logstash-Kibana.

###### **Why ELK stack is required?**

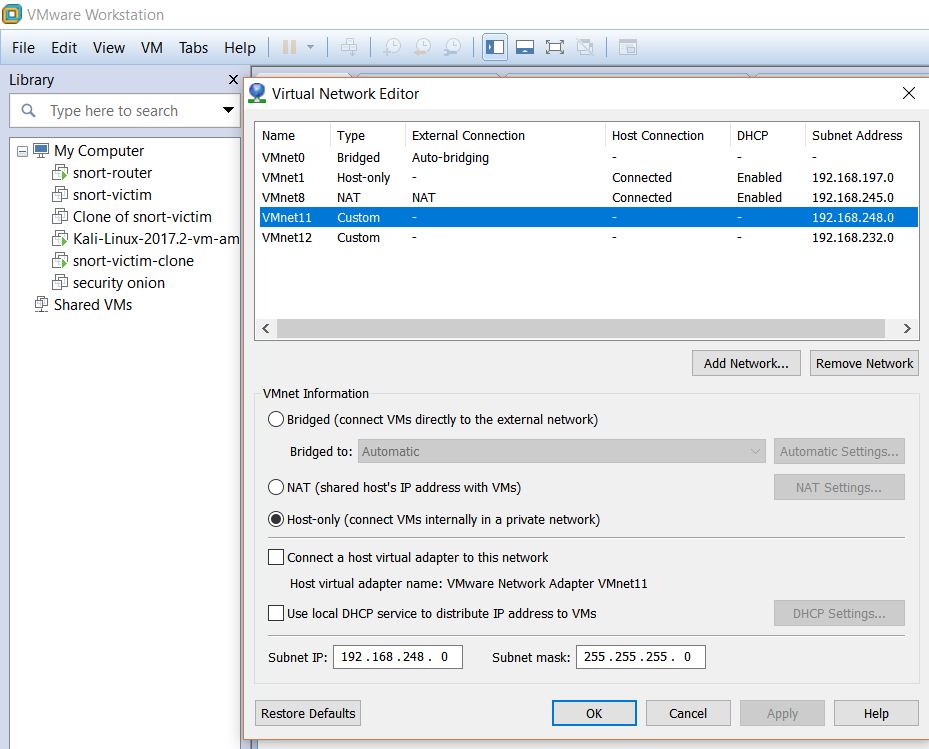
We use ELK stack because in this we do the analysis and pull the logs from the logstash and visualize it using kibana. This is an effective process and kibana has a high graphical user interface to visualize the logs.

# Experimental Setup

## Network setup

We have created a virtual network where all the machines, namely snort-router, attacker and victim are connected.

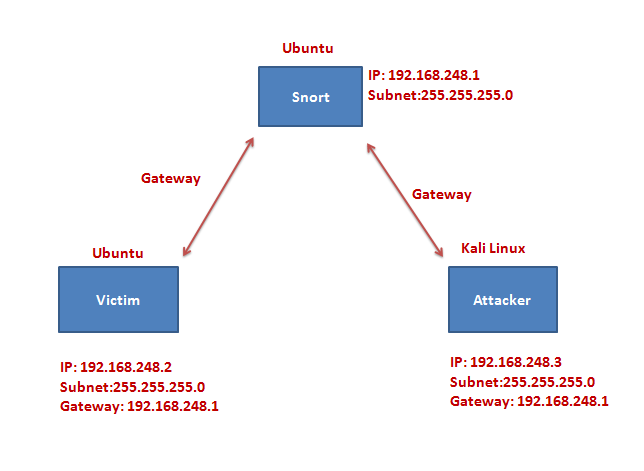
To create the virtual network, we should open VMware application and go to Edit -> Virtual Network Editor as follows:



In Virtual Network Editor, click on Add Network button and create the virtual network as shown in above figure. Once the virtual network is created we use this network to connect all the machines to this network.

## Virtual machines

The three virtual machines configuration what we have done look’s as follows:

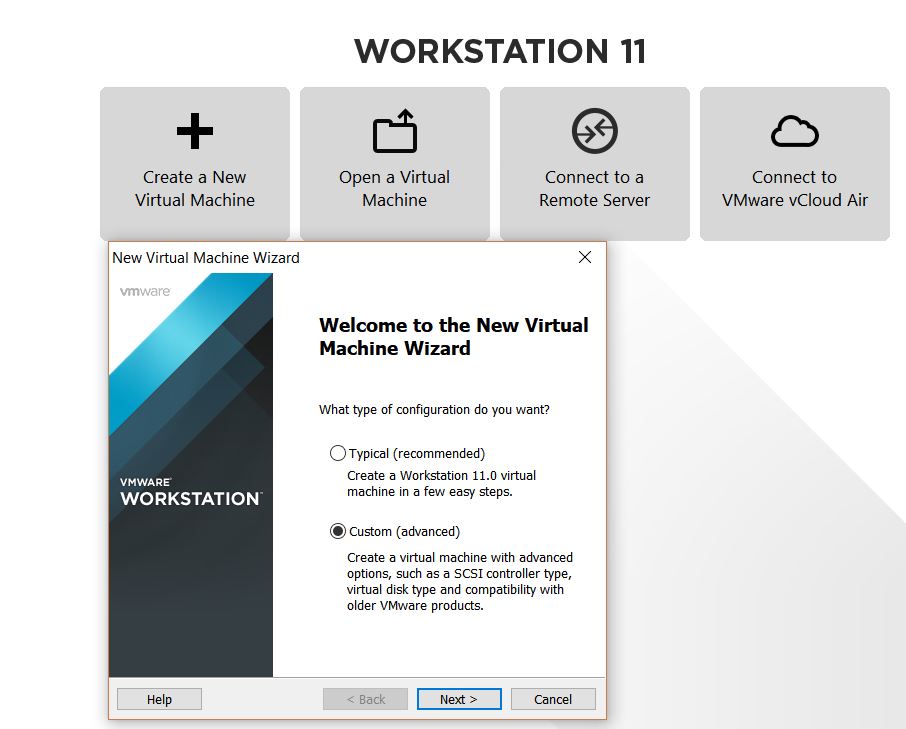


### Snort-router:

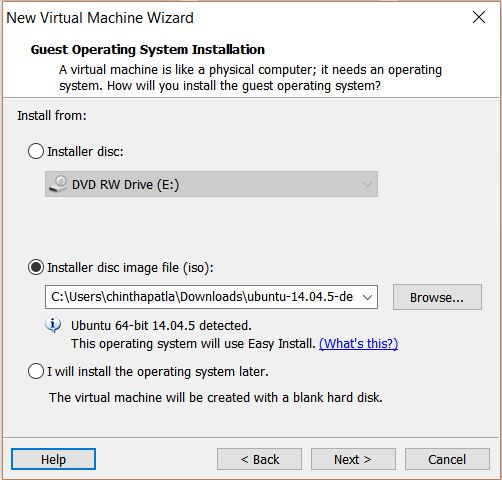
The router is a virtual machine running Ubuntu where snort along with all the required packages are installed. To begin we start installation of Ubuntu operating system as snort-router. The following screenshots explains this step.

#### Snort installation:

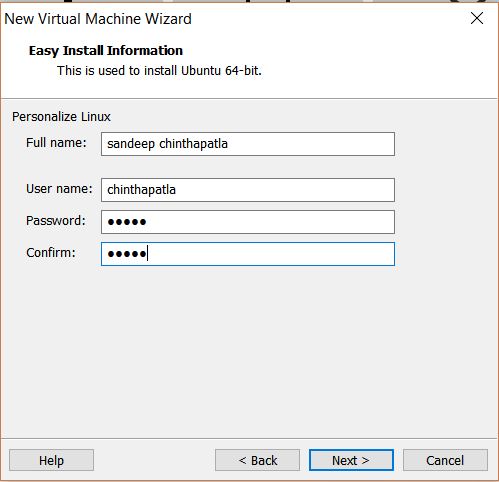
Step 1:



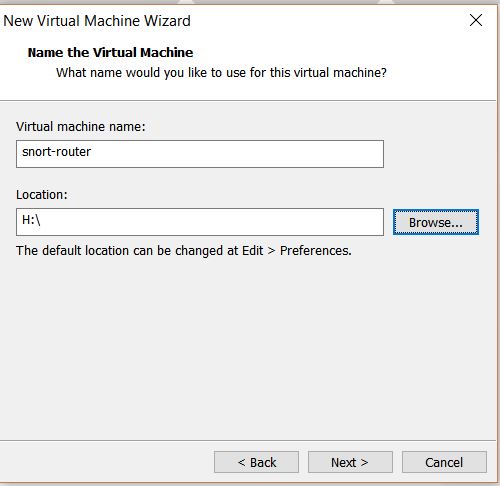
Step 2: We choose the Installer disc image file path as follows:



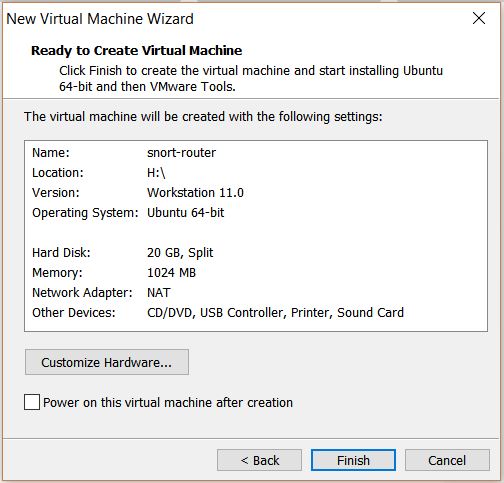
Step 3: We give the username and password as follows:



Step 4: In this step, we specify the virtual machine name

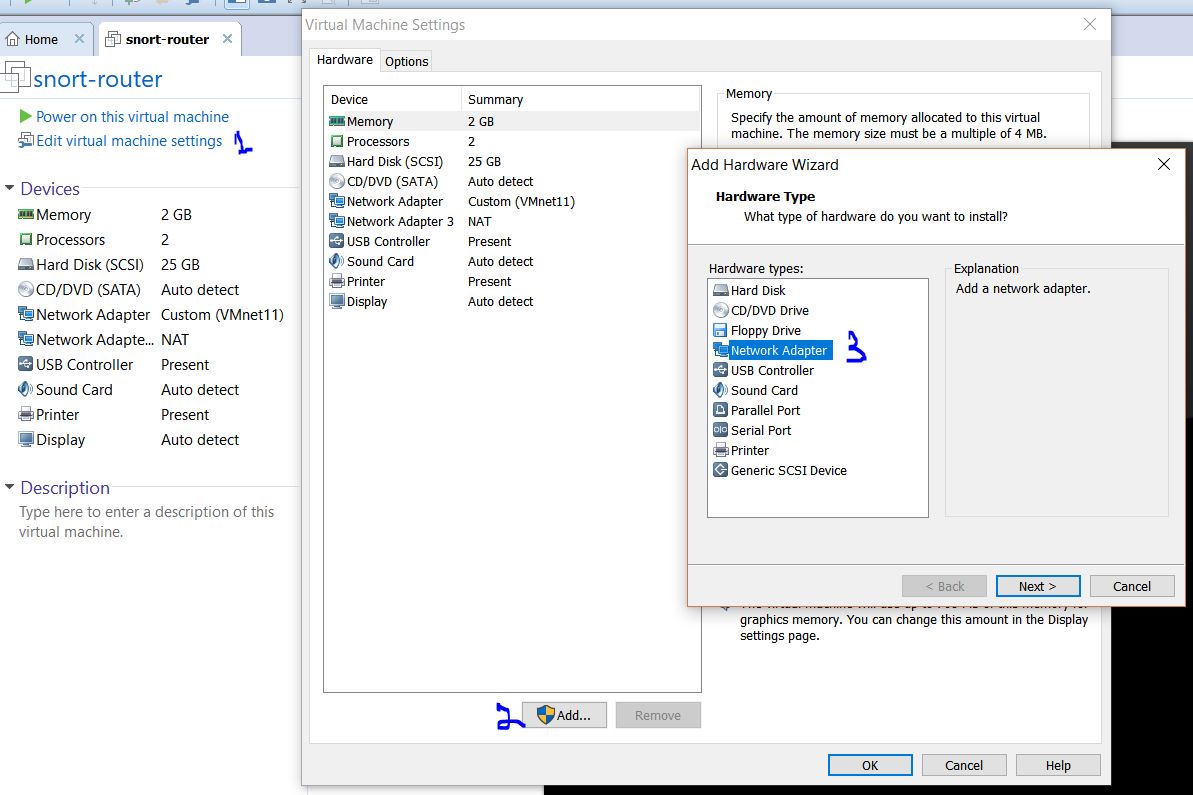


Step 5: This is the last step for creation of Virtual machine



Step 6: In this step, we follow sub-steps in creation of Network adapter as follows:

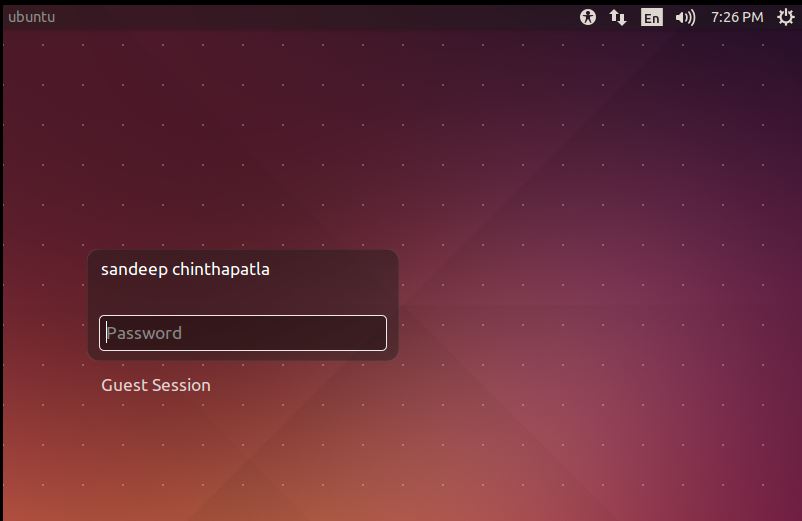
* + - 1. Go to edit Virtual machine settings
      2. Then click on Network adapter and then click on next



Step 7: Then create a Custom network as VMnet 11



Step 8: This screenshot shows that the Ubuntu has been installed successfully.



Now, after logging on we need go to terminal and install all the packages required for snort installation as follows:

Step 9: (XModulo, n.d.)

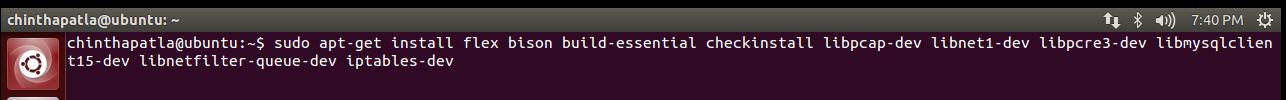
**build-essential**: install the build tools which are required to compile software.

**bison, flex:** Flex and Bison are Unix utilities that can help you to write parsers for random file formats

**libpcap-dev**: this a network traffic capture library required by Snort.

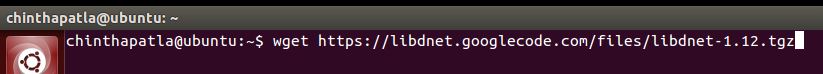
**libpcre3-dev**: Library of functions to support regular expressions required by Snort.

**libdumbnet-dev**: the libdnet library provides a simplified, portable interface to several low-level networking routines.



Step 10:

a) libdnet provides a simple, portable and network address manipulation. This package should be downloaded from libnet.googlecode.com

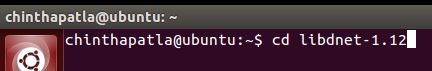


b) To extract the downloaded file, we type the command as

tar xvfvz libdnet-1.12.tgz and go to libdnet folder.

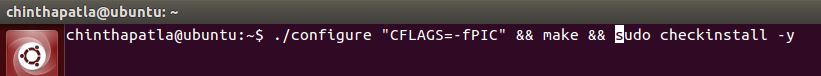


c) We change the directory as follows:



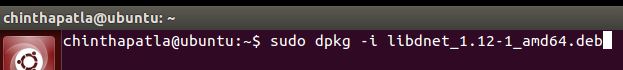
d) We do configuring as: -fPIC is required for 64-bit machines.

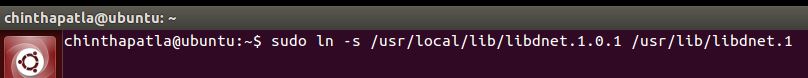
. /configure "CFLAGS=-fPIC"



e) We do the symbolic link creation as

sudo dpkg -i libdnet\_1.12-1\_amd64.deb





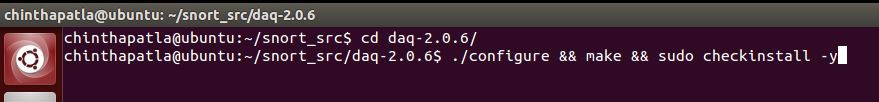
Step 11:

daq package: The DAQ creates an abstraction layer that allows snort to run on different hardware and software interfaces. (Snort FAQ, n.d.)

We install the daq package as,





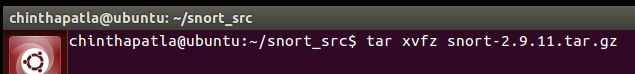


Step 12: Snort installation

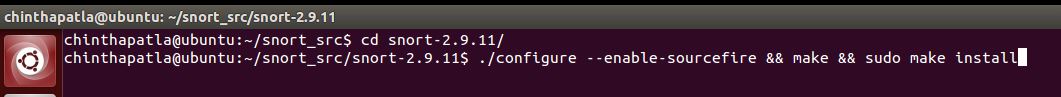
1. We create a snort directory and install as,



1. We extract the package as follows,



1. We then do the following step,

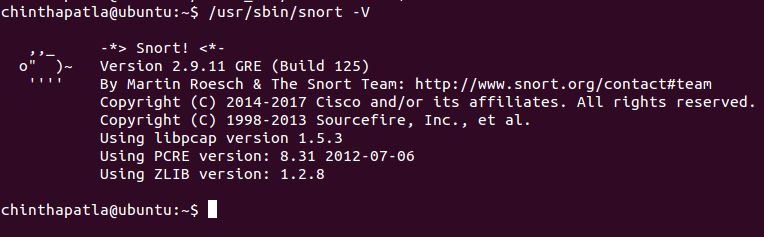


1. Ldconfig :- It updates all the libraries.





To check the version that is installed in our machine we do it as follows,



Step 13:

We create a group with name snort to run as unprivileged user.





Since Snort stores configuration files in **/etc/snort**, rules in **/etc/snort/rules**, **/usr/local/lib/snort\_dynamicrules**, and stores its logs in **/var/log/snort**, we copy the respective files to the specified folders.



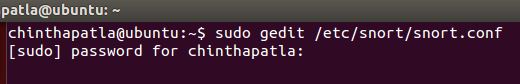






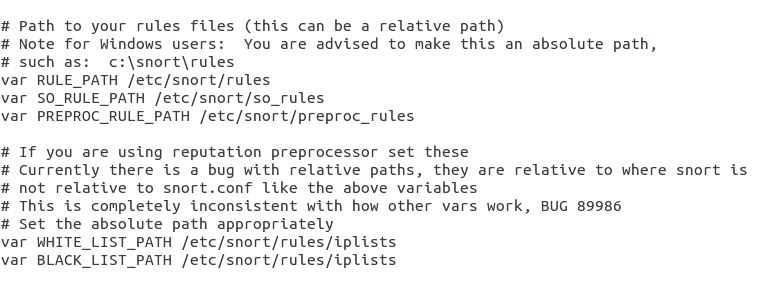
This command comments all the lines in the .config file that starts with include. This is done to exclude all the rule files that are used by Snort by default.

Now to run snort as NIDS we need to make some changes in the snort.conf file and also include our rule file path.



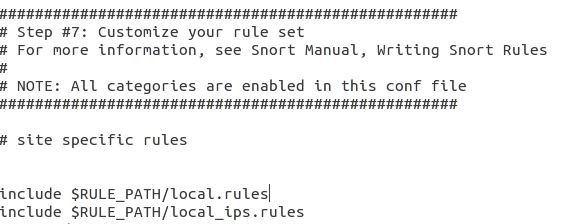
In this step, we specify the network that must be monitored by snort.



Next, we create the variables for our rule paths.

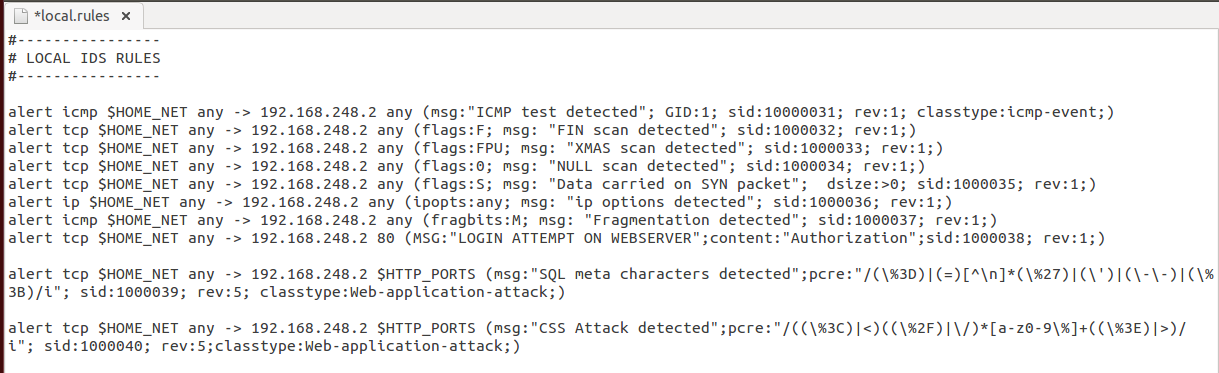
We have created two rules, namely local.rules and local\_ips.rules.

1. Local.rules -> in this we wrote all the alerts rules when snort runs in NID mode.
2. Local\_ips.rules -> in this we wrote all the alerts rules when snort runs in NIP mode

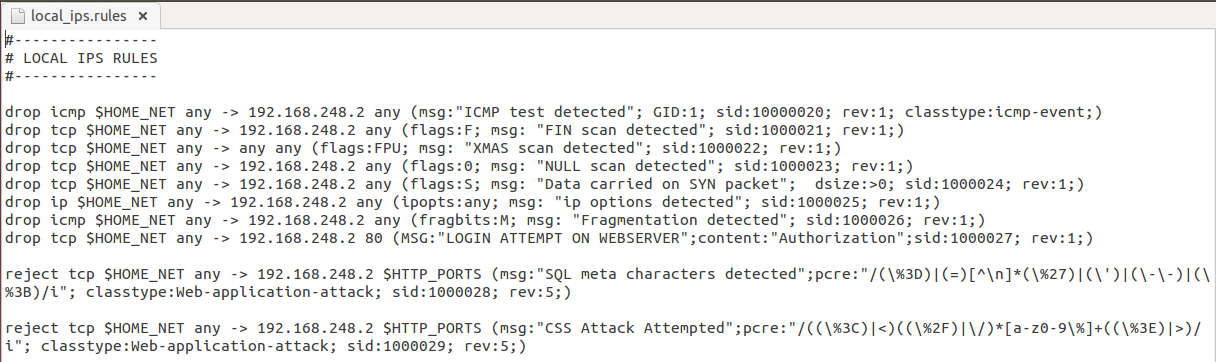


After changing the config file, we now open the local.rules and enter the alert rules for snort to detect.

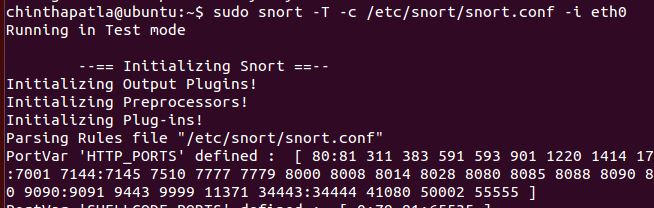
Rules for the snort to run as NIDS

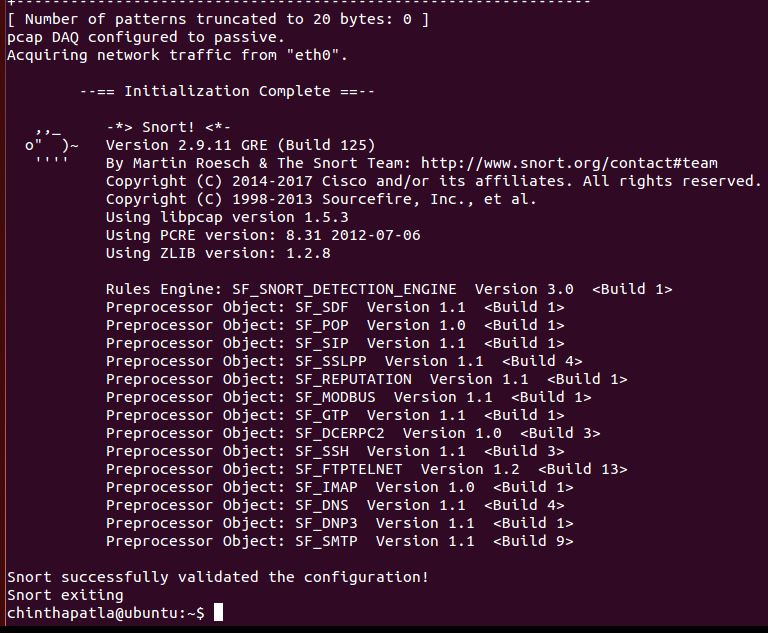


Rules for the snort to run as NIPS



After we make all the changes we need to test snort.conf file. We do it as follows:





Step 14:

Now we create the Snort directories:

sudo mkdir /etc/snort

sudo mkdir /etc/snort/rules

sudo mkdir /etc/snort/rules/iplists

sudo mkdir /etc/snort/preproc\_rules

sudo mkdir /usr/local/lib/snort\_dynamicrules

sudo mkdir /etc/snort/so\_rules

# Create some files that stores rules and ip lists

sudo touch /etc/snort/rules/iplists/black\_list.rules

sudo touch /etc/snort/rules/iplists/white\_list.rules

sudo touch /etc/snort/rules/local.rules

sudo touch /etc/snort/sid-msg.map

# Create our logging directories:

sudo mkdir /var/log/snort

sudo mkdir /var/log/snort/archived\_logs

# Adjust permissions:

sudo chmod -R 5775 /etc/snort

sudo chmod -R 5775 /var/log/snort

sudo chmod -R 5775 /var/log/snort/archived\_logs

sudo chmod -R 5775 /etc/snort/so\_rules

sudo chmod -R 5775 /usr/local/lib/snort\_dynamicrules

# Change Ownership on folders:

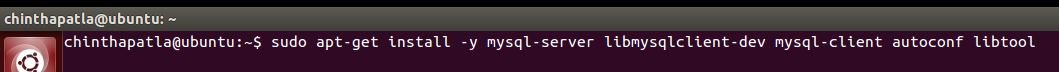
sudo chown -R snort:snort /etc/snort

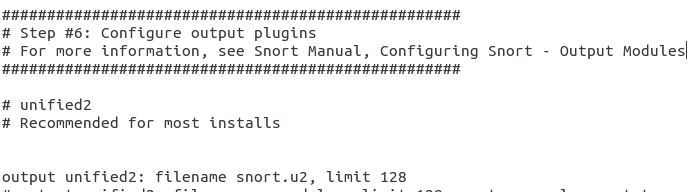
sudo chown -R snort:snort /var/log/snort

sudo chown -R snort:snort /usr/local/lib/snort\_dynamicrules

#### Barnyard2 installation:

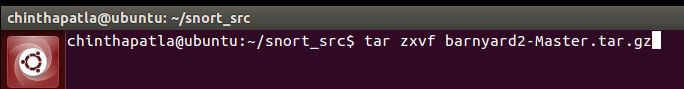
We install the pre-requisite which is MYSQL server as follows:

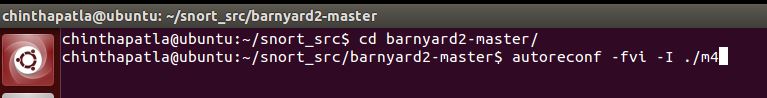






Now we extract by using the following command:





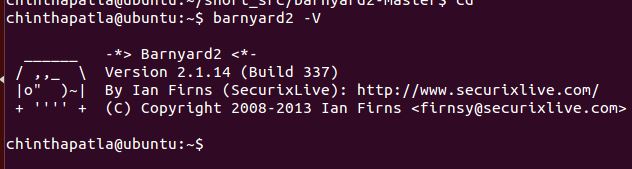












#### Creating MYSQL database:

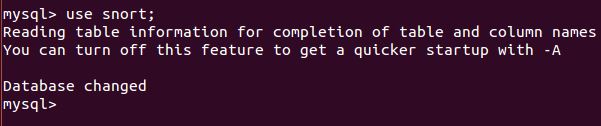
We create an account with password.



Then we create a database named “snort” to store all the log files



Then we use snort through the MYSQL



Then we create a schema in barnyard2:



Then we create a user named ‘snort-team’ identified by password as ‘snort’



Then we grant all the permissions to the user ‘snort-team’



Then we exit MYSQL



Now we edit the barnyard2.conf so that we can give barnyard2 the access to the database.

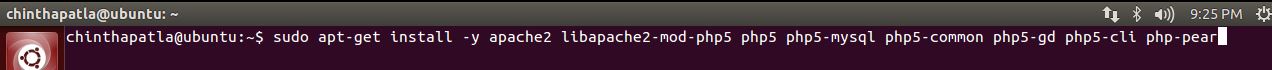




Then we change the permissions for barnyard2.



#### Base Installation:

We install BASE with the following pre-requisites:

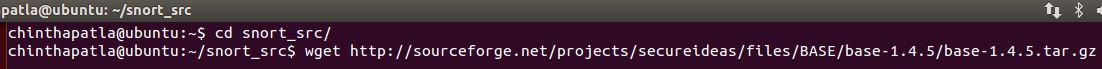






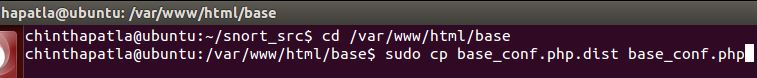






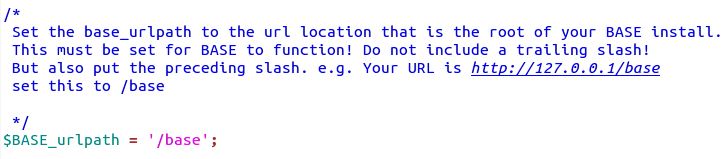


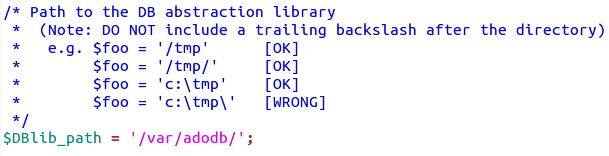


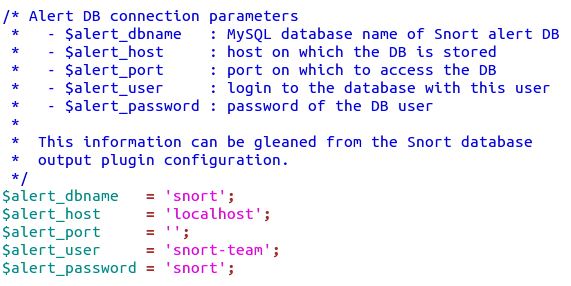


sudo vi /var/www/html/base/base\_conf.php

In the configuration file we make the following changes



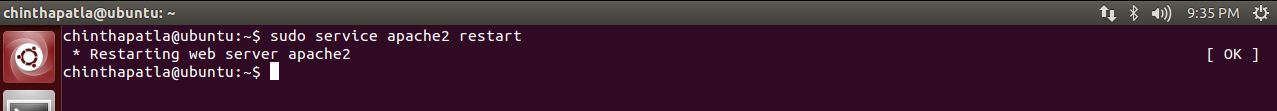






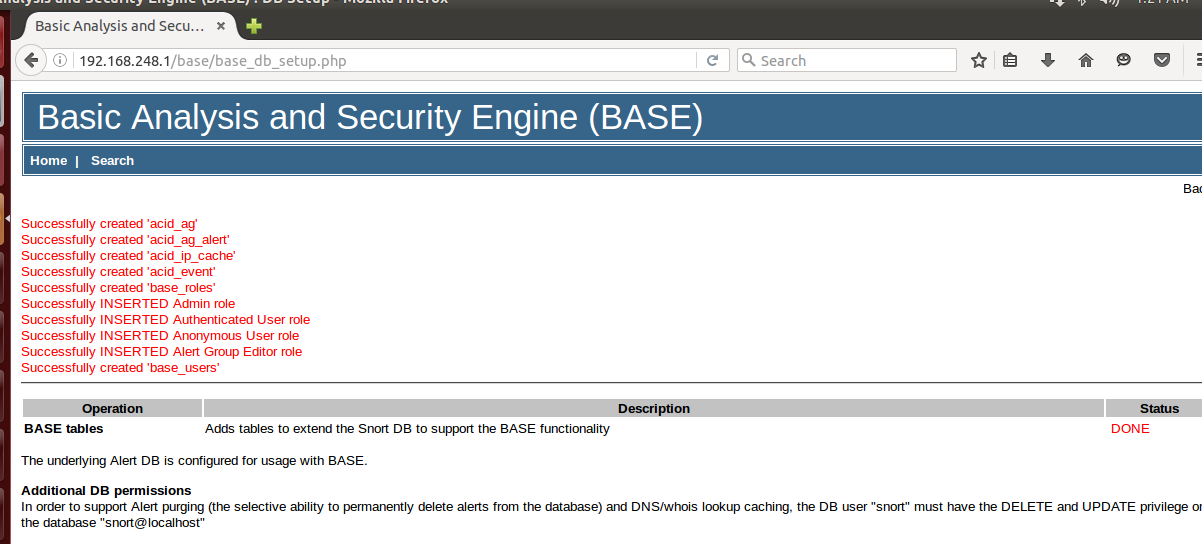


After changing the base\_conf.php we should restart the apache2 service



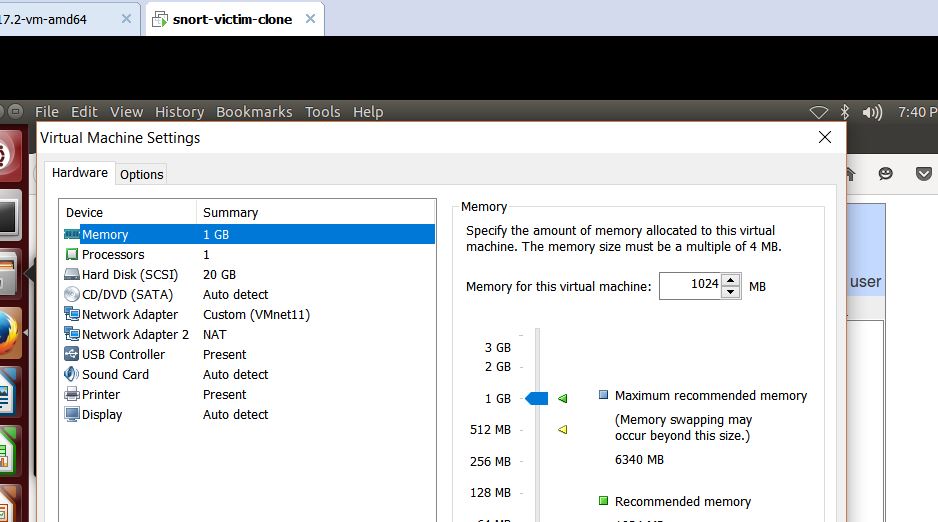
When we open the BASE page from the browser using [http://192.168.248.1:8080/BASE](http://192.168.248.1:8080/base).We need to click the button create BASE AG so that it creates some tables and insert some users to it.



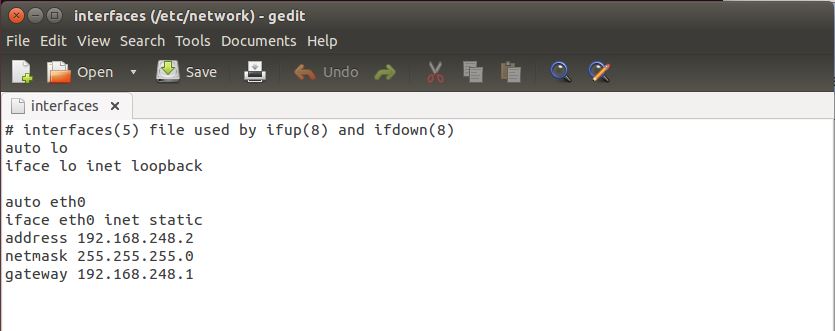


### Victim machine:

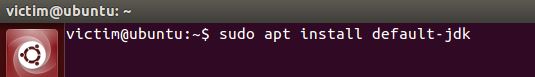
Now we make changes in the network interface of the Victim machine

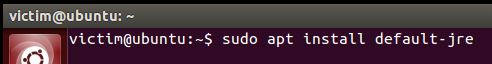


We go to sudo gedit /etc/network/interfaces and give the static eth0 interface



Now we install the pre-requisites for running Bodgeit store as:





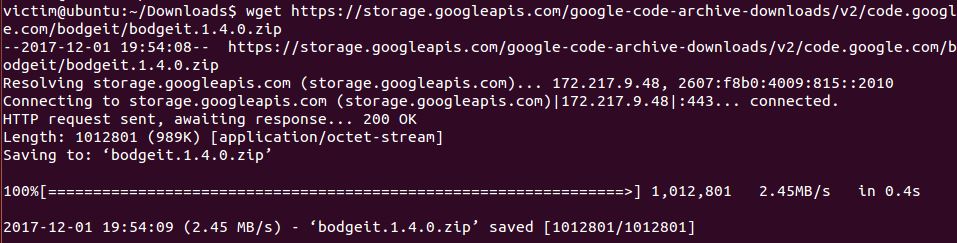


Then we change the path as,

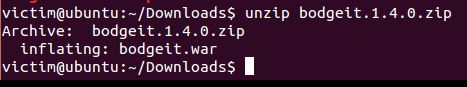


Now we download Bodgeit store as follows:





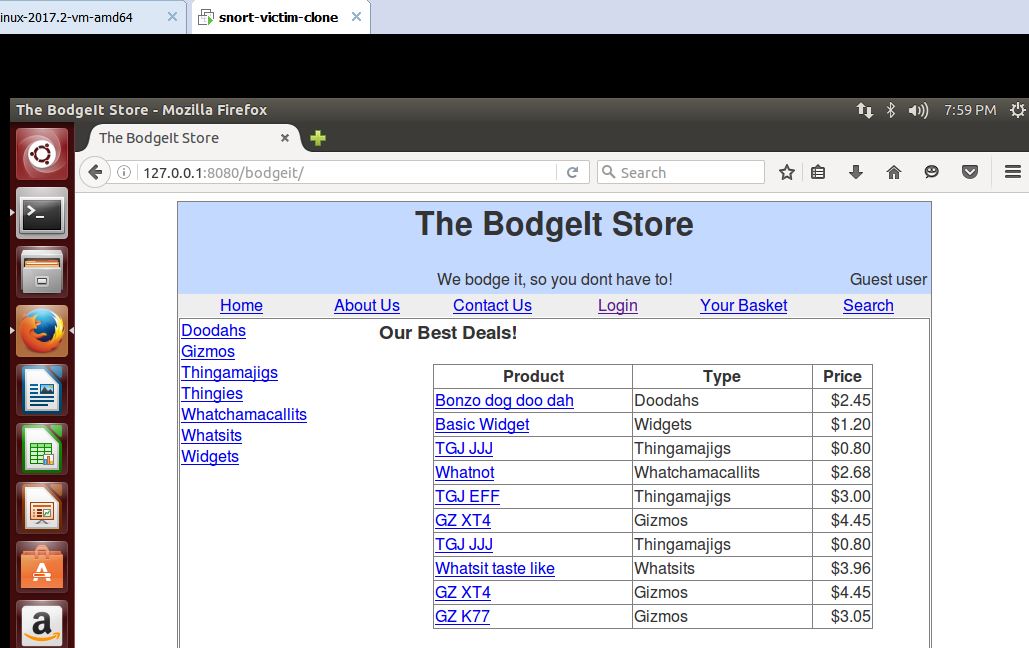
Then we unzip the file



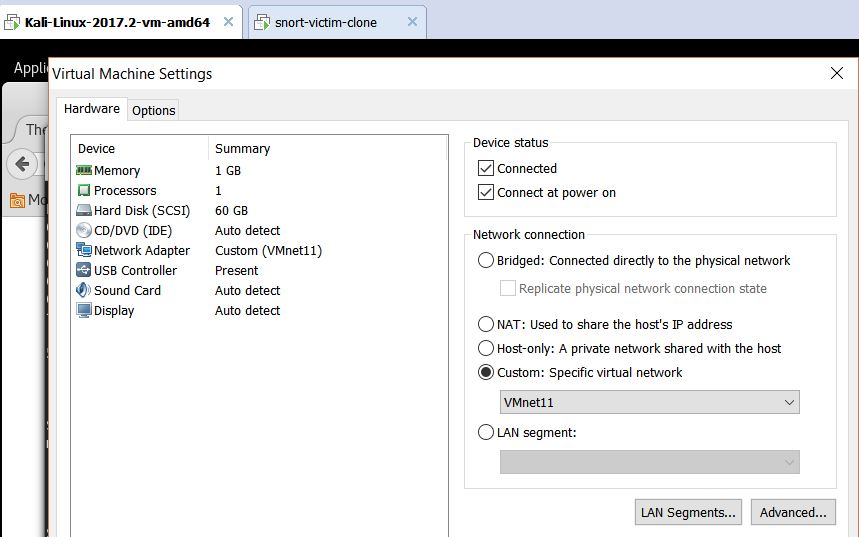
Then we move using the move command



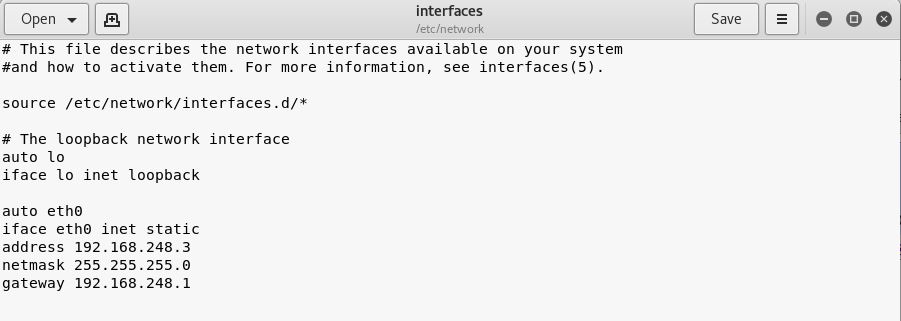
This is the preview of the Bodgeit store running in the Victim machine



### Attacker machine:



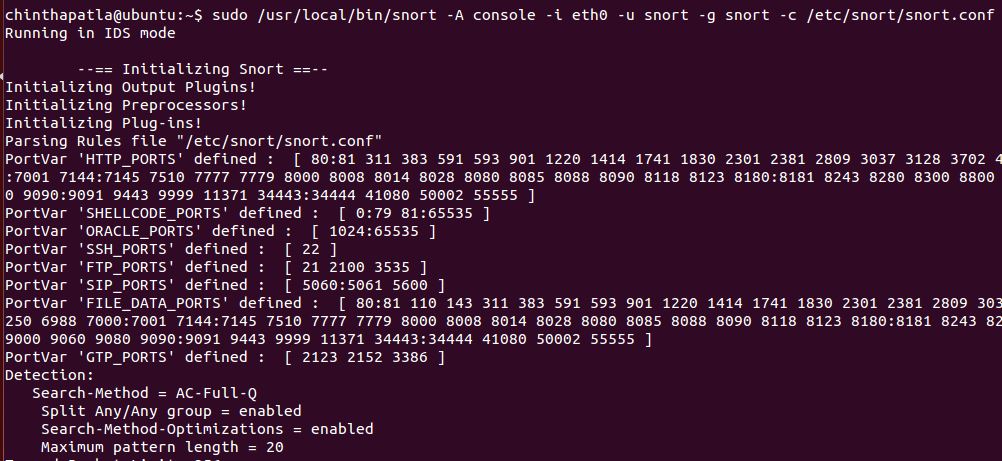
After starting the kali machine, we will open the interfaces file in /etc/network

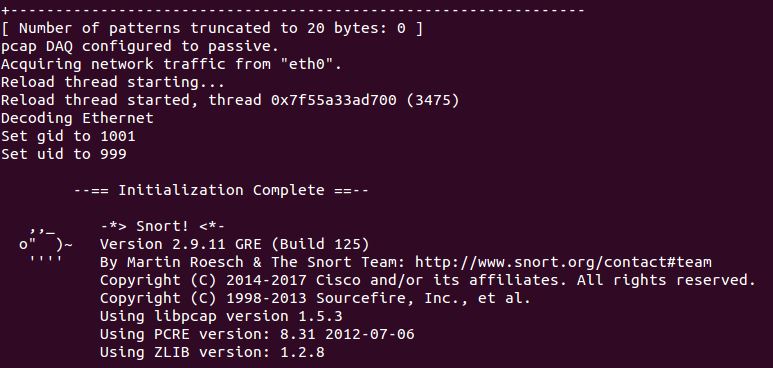


## Snort running as NIDS

#### Running snort in IDS mode

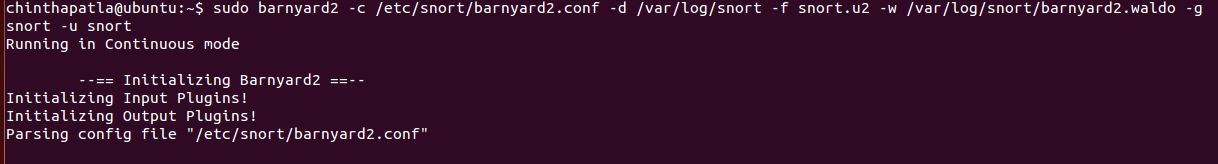
sudo /usr/local/bin/snort -A console -i eth0 -u snort -g snort -c /etc/snort/snort.conf

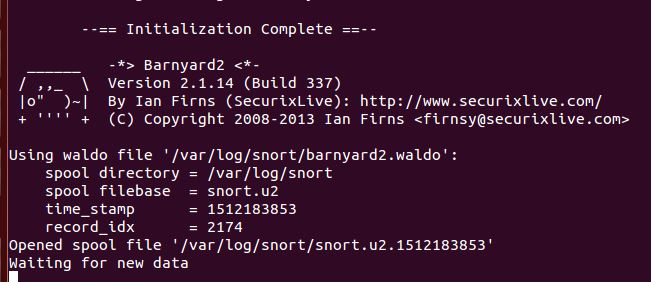




#### Running baryard2

In another terminal

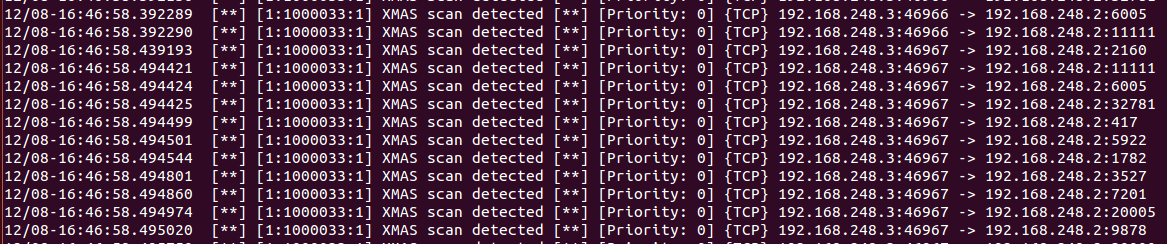




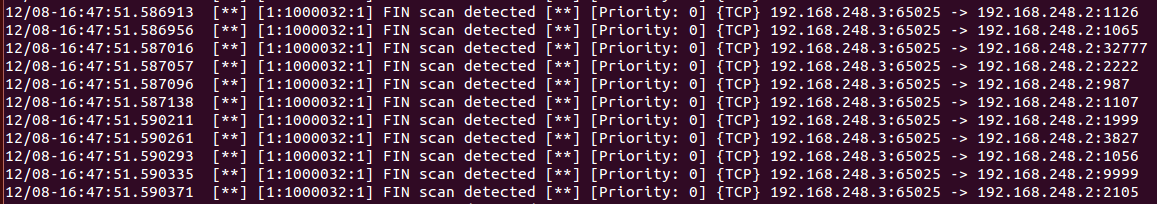
#### Results:

Once the attacks are performed from the attacker then we can see the snort alerts on the snort-router. We have performed various scans from the attacker on the victim machine and noticed several alerts on the machine where snort is running.

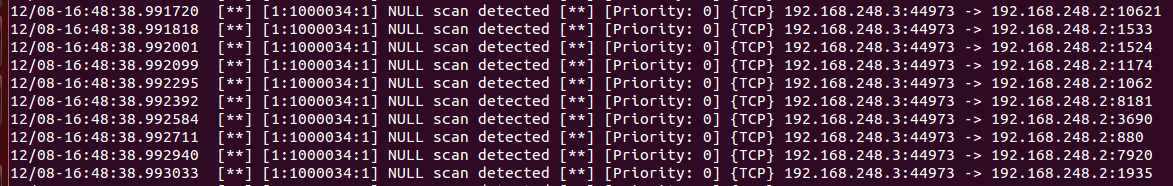
When we performed XMAS scan below alert is popped up.



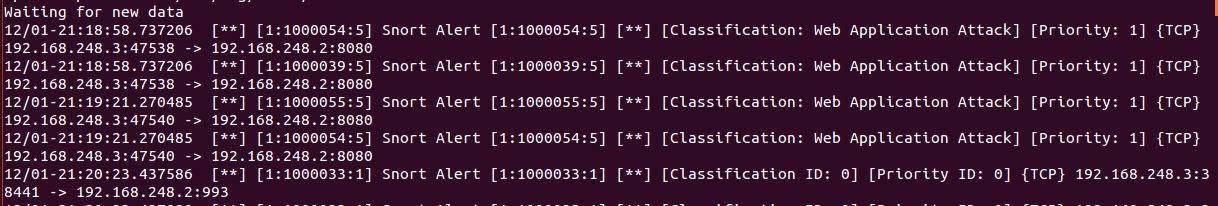
When we performed FIN scan below alert is popped up.



When we performed NULL scan below alert is popped up.



When we perform SQL and XSS attacks we can see the below alerts in barnyard2 as Snort Alerts.



**Attacks:**

**SQL Injection Attack:**

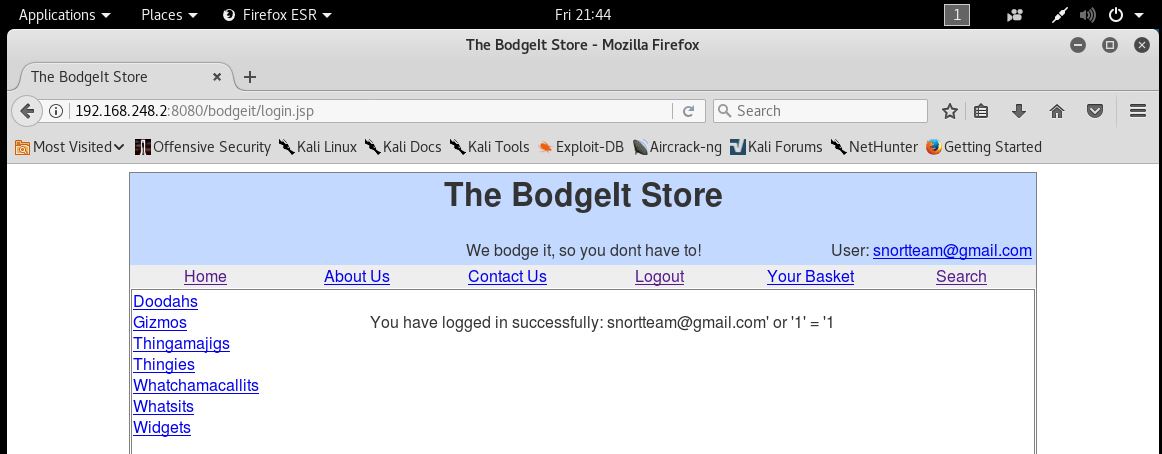
It isa code injection attack where attacker tries to attack backend database connected to the application by inserting malformed SQL statement.  Intruders usually take advantage of poorly designed codes mainly the web application to spread the malicious codes to exploit information.

(IEEEXPLORE, n.d.) (owasp, n.d.)

**Performing SQL injection attack:**

Now the SQL injection attack has been performed from the attacker machine as:

Username’ or ‘1’ =’1

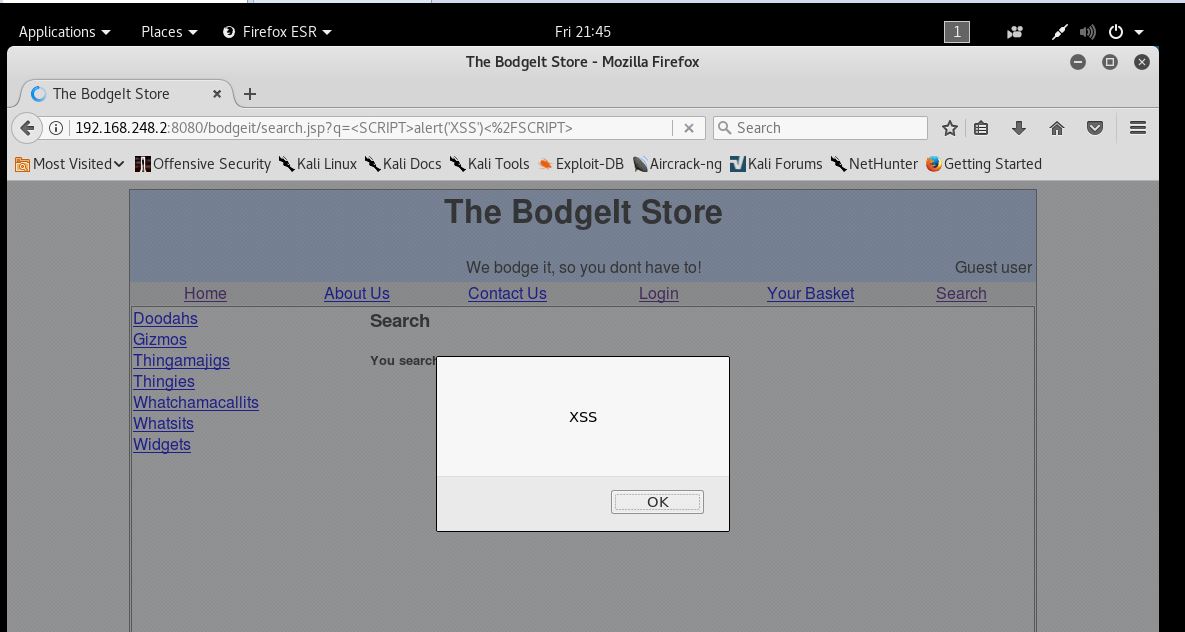


**Cross-site scripting** (**XSS**): “It is a type of [computer security](https://en.wikipedia.org/wiki/Computer_security) [vulnerability](https://en.wikipedia.org/wiki/Vulnerability_(computer_science)) typically found in [web applications](https://en.wikipedia.org/wiki/Web_application). XSS enables attackers to [inject](https://en.wikipedia.org/wiki/Code_injection) [client-side scripts](https://en.wikipedia.org/wiki/Client-side_script) into [web pages](https://en.wikipedia.org/wiki/Web_page) viewed by other users. A cross-site scripting vulnerability may be used by attackers to bypass [access controls](https://en.wikipedia.org/wiki/Access_control) such as the [same-origin policy](https://en.wikipedia.org/wiki/Same-origin_policy).” (wikipedia, n.d.)

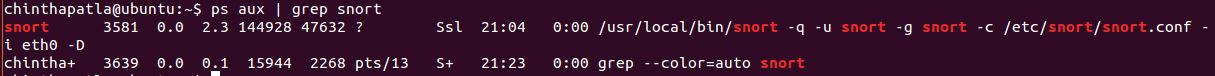
**Perfroming XSS attack:**

Cross Site Scripting attack has been performed from the attacker machine as follows:

<script>alert(“XSS”)</script>

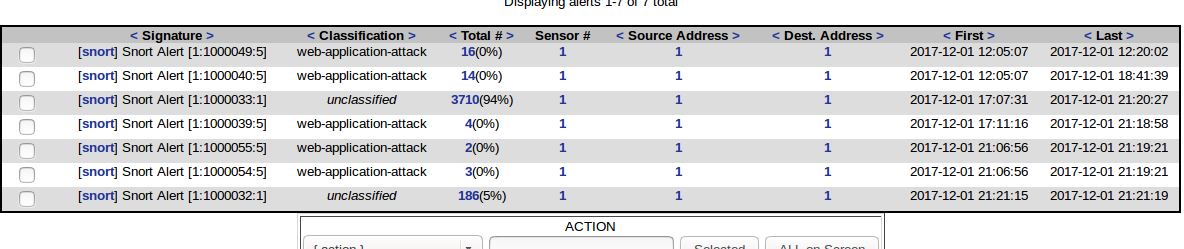


After performing all the attacks, we must kill the snort process running in the daemon mode.



We analyze the traffic of the network using BASE as:

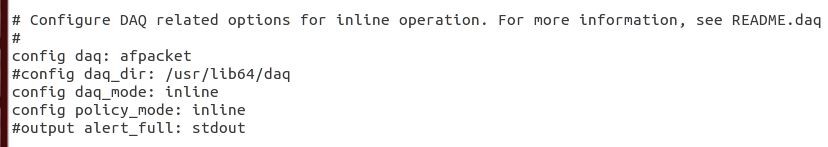




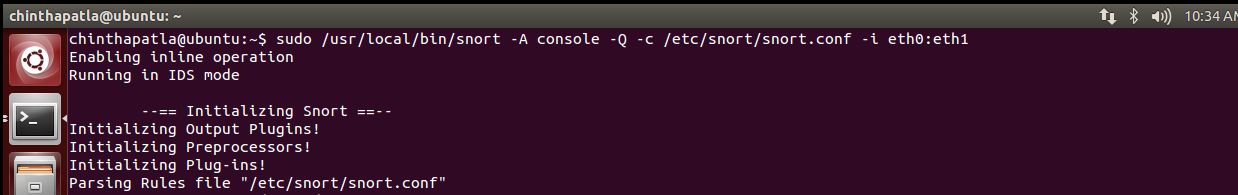
## Snort running as NIPS

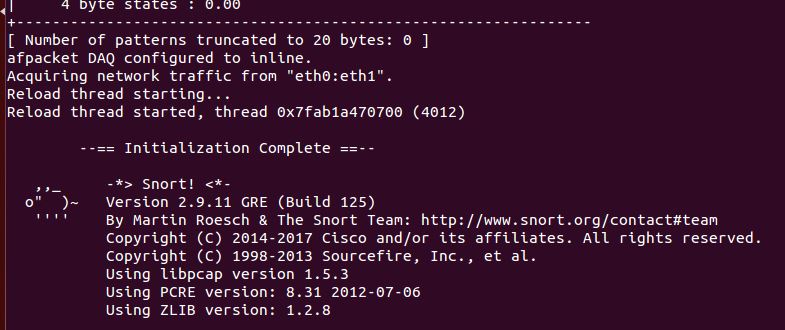
#### Running snort in IPS mode

To run snort in IPS mode we need to uncomment the following lines in /etc/snort/snort.conf



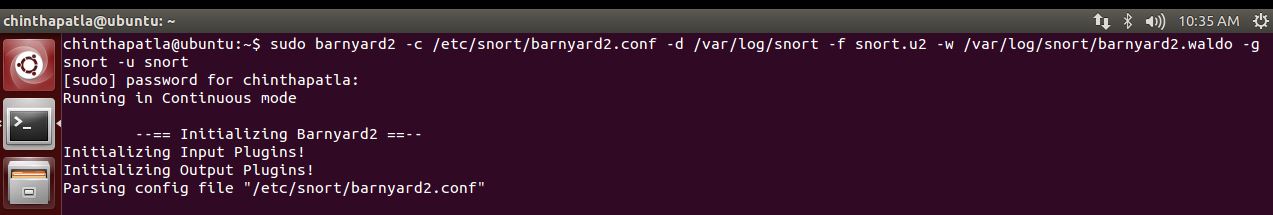
sudo /usr/local/bin/snort -A console -Q -c /etc/snort/snort.conf -i eth0:eth1

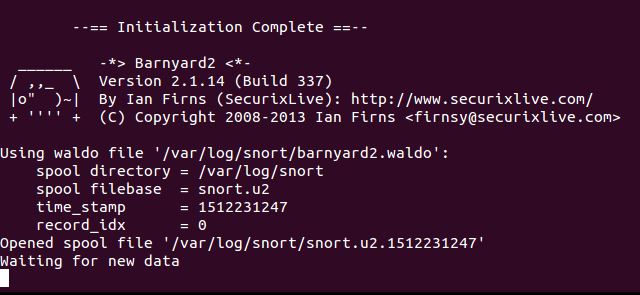




#### Running Barnyard2:

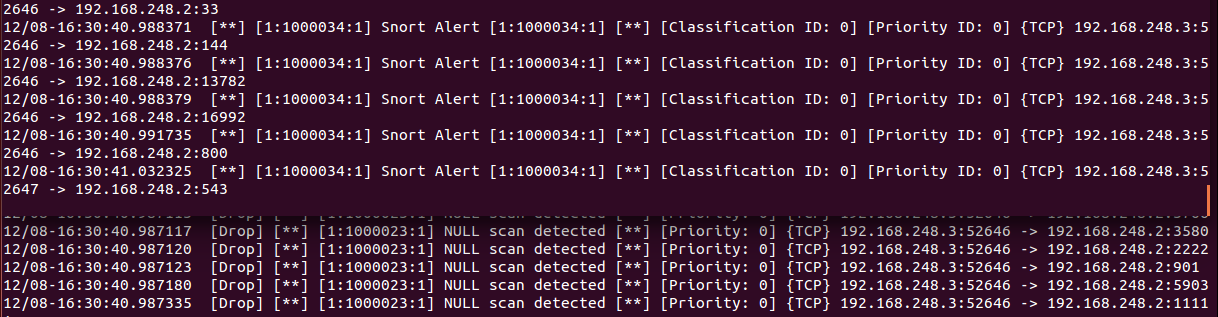
sudo barnyard2 -c /etc/snort/barnyard2.conf -d /var/log/snort -f snort.u2 -w /var/log/snort/barnyard2.waldo -g snort -u snort



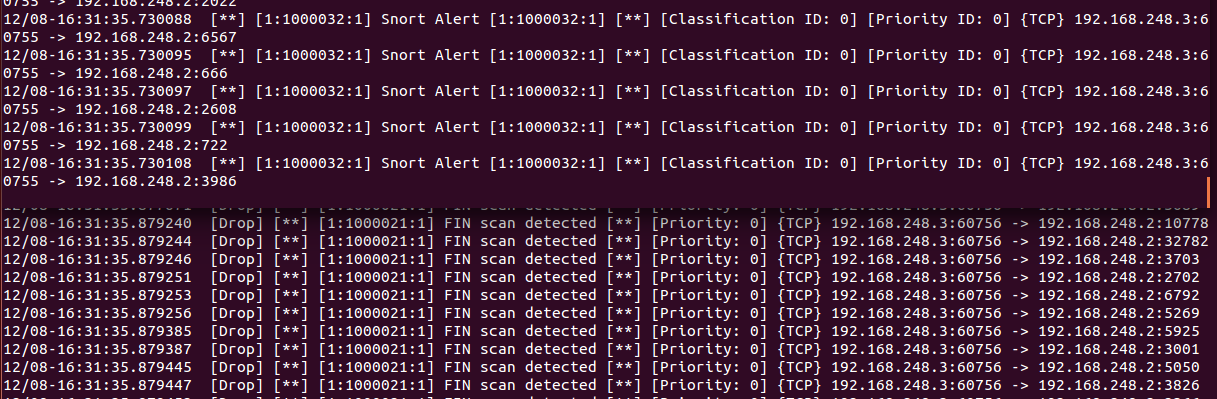


#### Results:

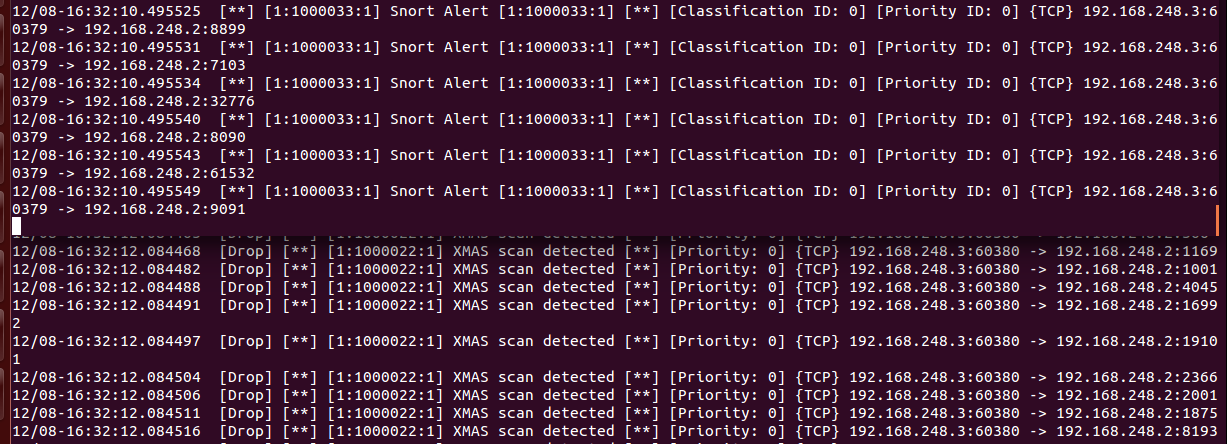
When we performed NULL scan, we can see packets being dropped and also see the events being notices in barnyard2 while snort running in inline mode.



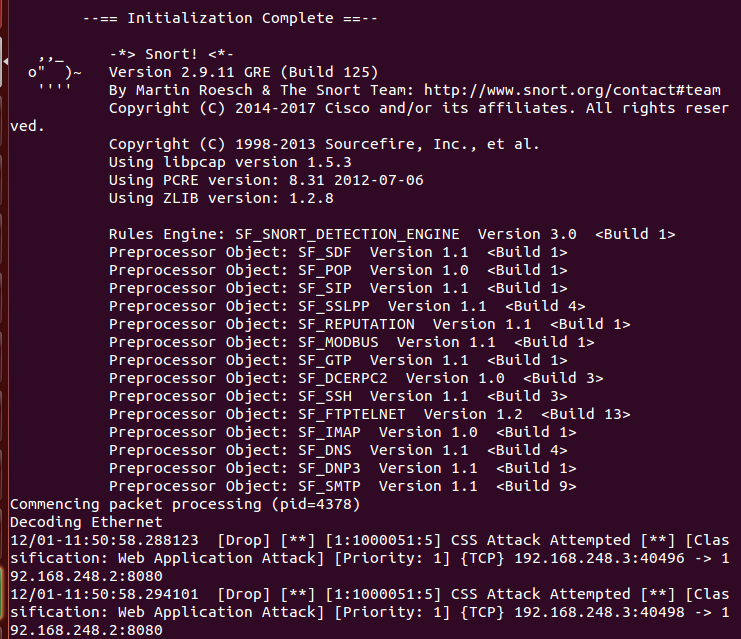
When we performed FIN scan, we can see packets being dropped and also see the events being notices in barnyard2 while snort running in inline mode.



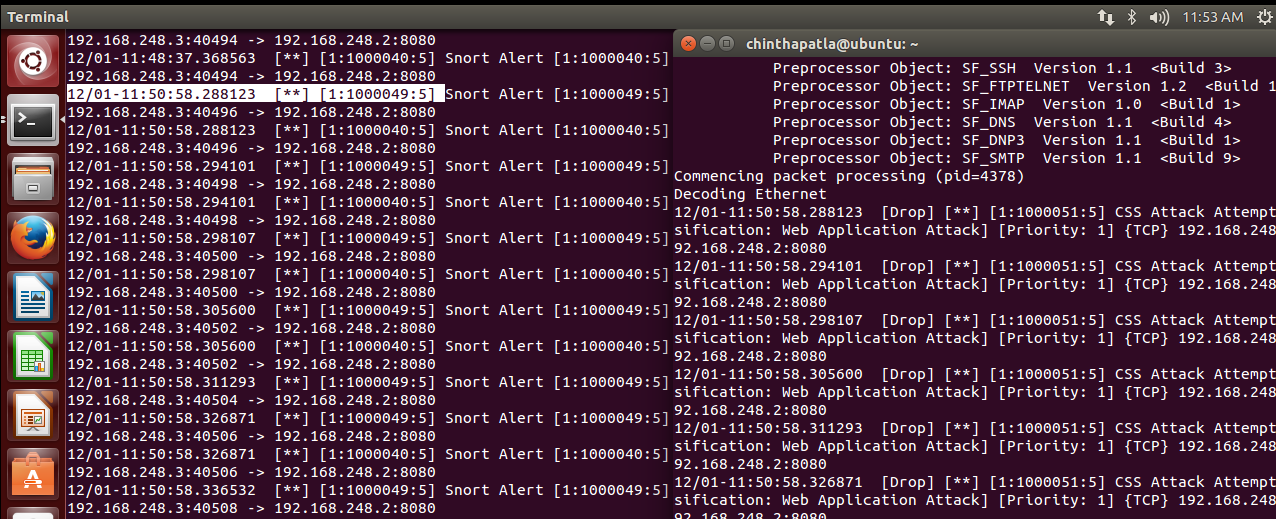
When we performed XMAS scan, we can see packets being dropped and also see the events being notices in barnyard2 while snort running in inline mode.



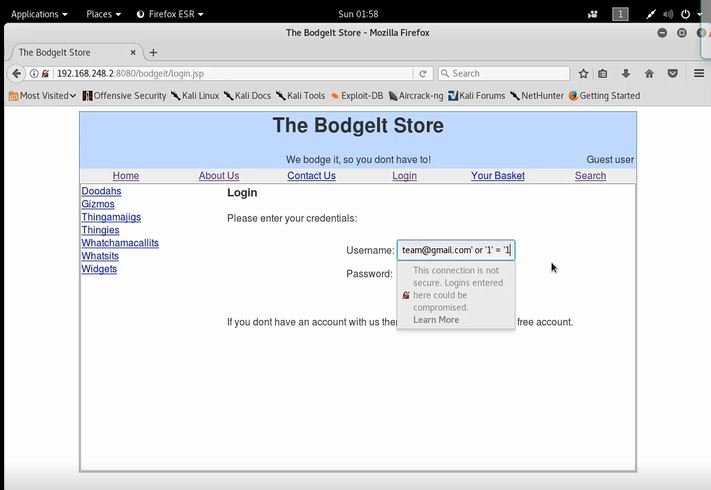
We can see that the packets are dropped when the attacks are performed by the attacker.

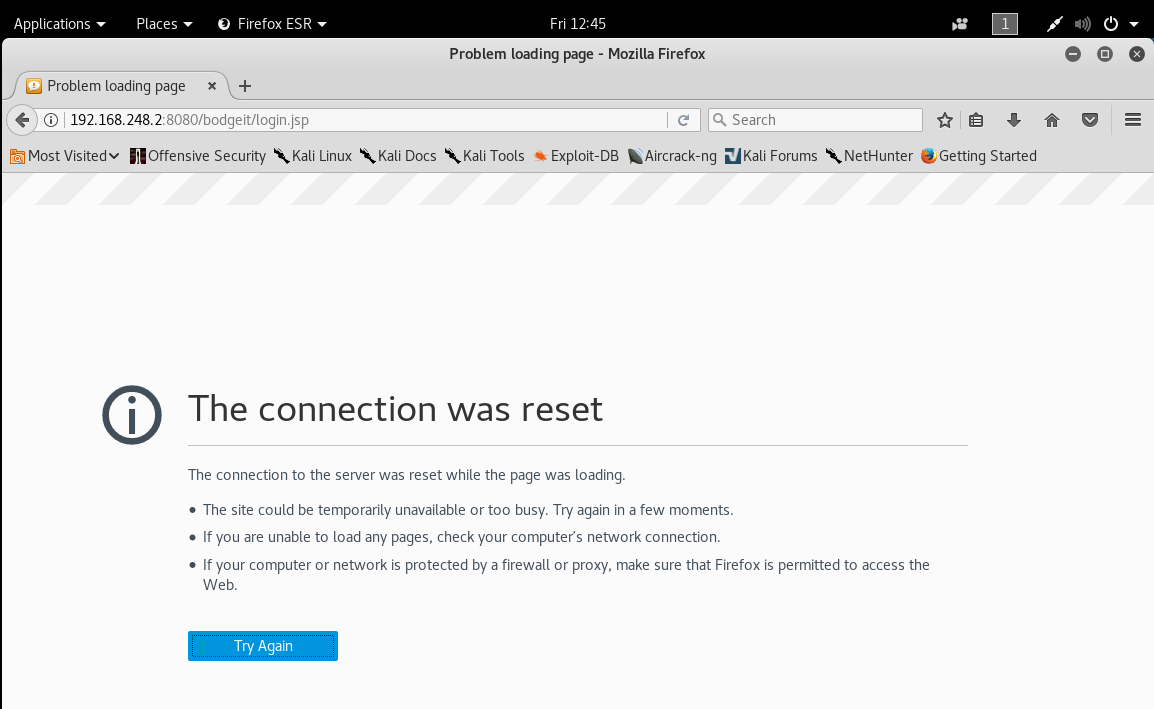






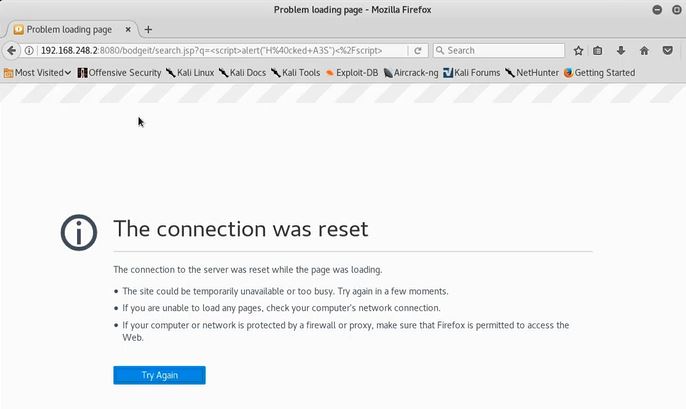
When the attacker tries to perform SQL injection attack the below page appears.



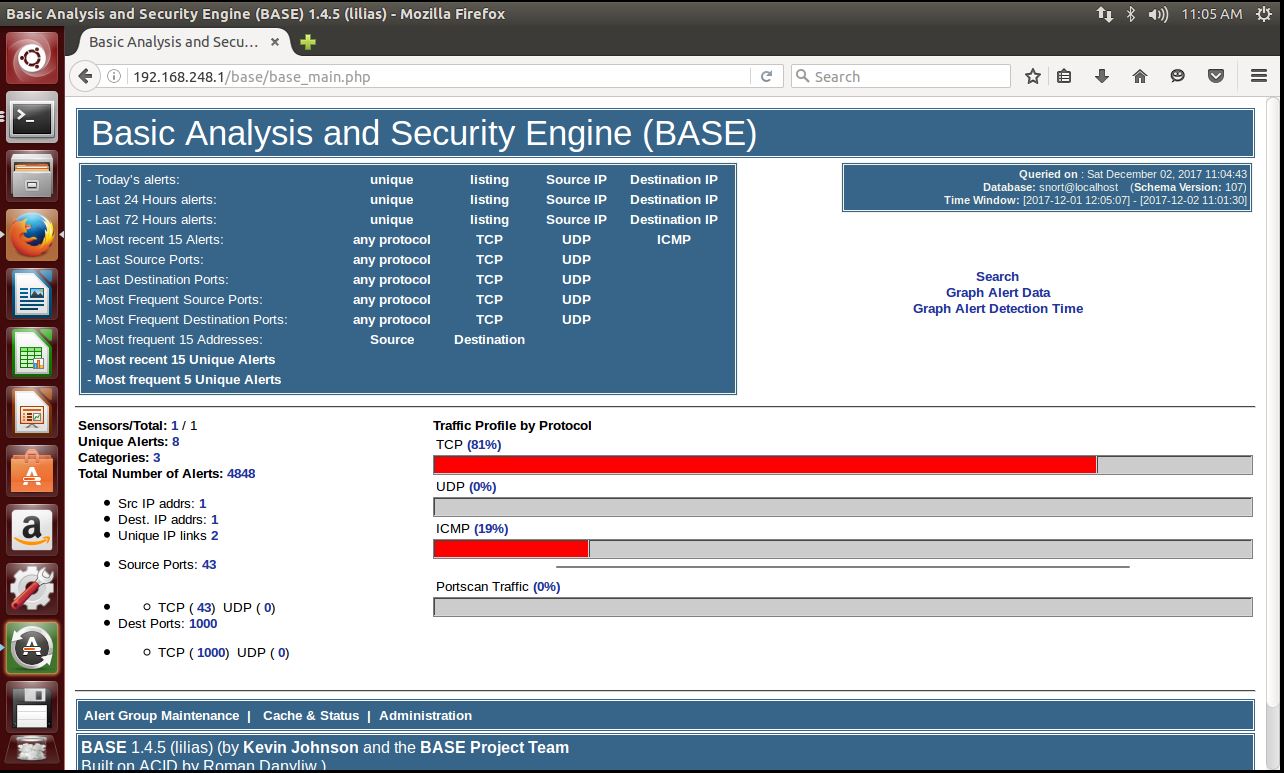


When the attacker tries to perform Cross Site Scripting attack the below page appears.

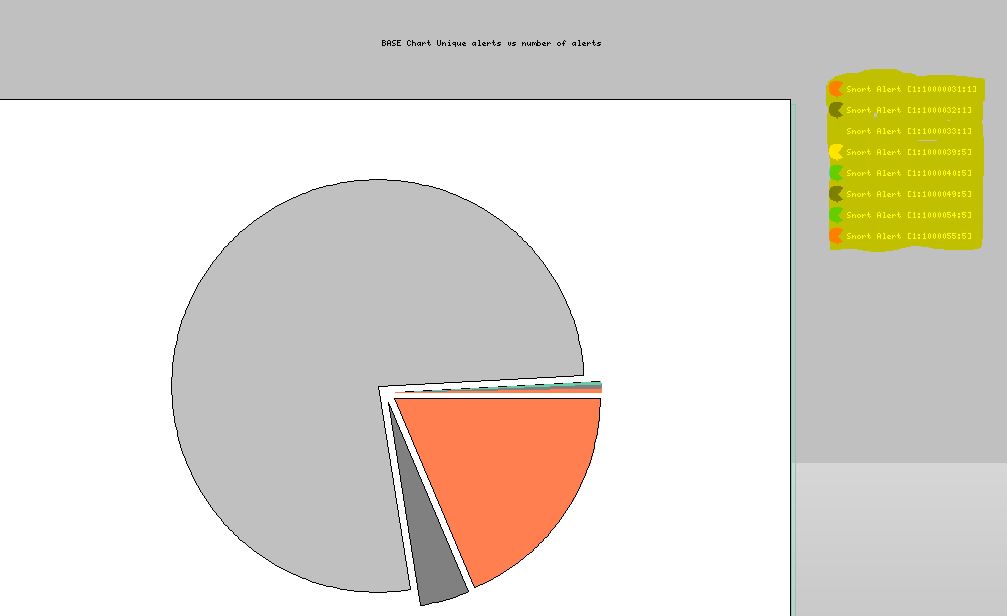




We can see the analysis of the attacks being performed and the unique alerts in the BASE.



We can also plot the alerts recorded in form of pie-chart.



**Challenges:**

While doing this project we faced several challenges but as a team we kept on going by solving each one of them. But the major challenge we faced is to feed the kibana with json logs. This is because Snort cannot produce json format logs and we could not find any solution to convert the unified2 format logs generated by Snort to json format.

# Conclusion and Learned skills

In this project we all have experienced to understand in depth how the Snort IDPS works like installing, implementing, testing and analyzing while detecting a malicious activity mainly SQL injection and Cross side scripting.

# Future work

As future work, we intend to implement more attacks in Snort because we performed only two attacks (Sql injections and Cross site scripting). We will also be expecting to solve our current existing issues for changing the snort unified2 packet into JSON format, to run the elasticsearch, logstash and kibana successfully.

# References

*Apache tomcat*. (n.d.). From http://tomcat.apache.org

*exciting ip*. (n.d.). From http://www.excitingip.com/636/snort-open-source-intrusion-detection-system/

*Generating artificial snort alerts and implementing SELK:The Snort-elastic-logstash-kibana.* (2017, Sep). From (http://www.arl.army.mil/arlreports/2017/ARL-TR-8175.pdf)

*IEEEXPLORE*. (n.d.). From ttp://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7053873

*Infosec*. (n.d.). From http://resources.infosecinstitute.com/the-bodgeit-store-part-1-2/#gref

*libdnet*. (n.d.). From libdnet.sourceforge.net

*NIST*. (n.d.). From Intrusion detection and prevention systems: (http://ws680.nist.gov/publication/get\_pdf.cfm?pub\_id=901146)

*owasp*. (n.d.). From https://www.owasp.org/index.php/SQL\_Injection

*OWASP*. (n.d.). From https://www.owasp.org/index.php/SQL\_Injection

*Snort FAQ*. (n.d.). From Snort.org: https://www.snort.org/faq/readme-daq

*Snort manual*. (n.d.). From http://manual-snort-org.s3-website-us-east-1.amazonaws.com/node5.html

*Snort-org*. (n.d.). From https://www.snort.org/

*Talos intelligence*. (n.d.). From https://www.talosintelligence.com/snort

*Tech target*. (n.d.). From Tech target: http://searchservervirtualization.techtarget.com/definition/virtual-machine

*Ubuntu*. (n.d.). From https://help.ubuntu.com/lts/installation-guide/s390x/ch01s01.html

*wikipedia*. (n.d.). From https://en.wikipedia.org/wiki/Java\_(programming\_language)

*wikipedia*. (n.d.). From https://en.wikipedia.org/wiki/PHP

*wikipedia*. (n.d.). From https://en.wikipedia.org/wiki/Cross-site\_scripting

*Wikipedia*. (n.d.). From Intrusion detection system: (https://en.wikipedia.org/wiki/Intrusion\_detection\_system)

*Wikipedia*. (n.d.). From Kali-Linux: https://en.wikipedia.org/wiki/Kali\_Linux

*Wikipedia*. (n.d.). From https://en.wikipedia.org/wiki/Ubuntu\_(operating\_system)

*Wikipedia*. (n.d.). From (https://en.wikipedia.org/wiki/MySQL)

*Wikipedia*. (n.d.). From https://en.wikipedia.org/wiki/Cross-site\_scripting)

*Wikipedia*. (n.d.). From https://en.wikipedia.org/wiki/Snort\_(software)

*XModulo*. (n.d.). From http://xmodulo.com/how-to-compile-and-install-snort-from-source-code-on-ubuntu.html