Cyber Security Project IDS

Name: Muhammad Muneeb
Roll # 7718

Step no 1:

I have installed snort ids on linux using following command.

sudo apt install snort

This command installed snort on my local linux machine. During installation it asked for my subnet mask which I found using the following command.

ip a s

After that the intallion proceeded swiftly without any issues.

Step no 2: (Configure Snort for Network Monitoring)

After that I analyzed the snort.conf file present in this location.

/etc/snort/snort.conf

Since most of the snort functionality is solely based on rules so in this term this file is the most important one.

On the top section we can configure variable for HOME_NET and we can configure it according to our subnet so that snort may analyze all machines present on that subnet.

Furthermore, we can see one more variable namely RULE_SET and this varisable is set to rules directory.

/etc/snort/rules

That means snort is loading rules from this directory by default.

After that we can see that there is hiererchy which is maintained in this conf file and around 8 steps are mensioned in this file.

But since we are interested in rule we jumped to step 7 of this hiererchy and there we can see all the active rules file which are present in rule directory mentioned above. we can easily disable all the predefined community rules using vim. And we

can only leave our local.rule file so that we can test our custom rules for testing purpose.

Step 3: (Create Snort Rules)

For checking purpose I have disabled all community rules except for my local rules.

```
#
# If you install the official VRT Sourcefire rules please review this
# configuration file and re-enable (remove the comment in the first line) those
# rules files that are available in your system (in the /etc/snort/rules
# directory)

# site specific rules
include $RULE_PATH/local.rules

# The include files commented below have been disabled
# because they are not available in the stock Debian
# rules. If you install the Sourcefire VRT please make
# sure you re-enable them again:

#include $RULE_PATH/app-detect.rules
#include $RULE_PATH/attack-responses.rules
#include $RULE_PATH/backdoor.rules
#include $RULE_PATH/bad-traffic.rules
##include $RULE_PATH/blacklist.rules
##include $RULE_PATH/botnet-cnc.rules
##include $RULE_PATH/browser-chrome.rules
##include $RULE_PATH/browser-firefox.rules
##include $RULE_PATH/browser-ie.rules
##include $RULE_PATH/browser-other.rules
##include $RULE_PATH/browser-plugins.rules
##include $RULE_PATH/browser-webkit.rules
#include $RULE_PATH/chat.rules
##include $RULE_PATH/content-replace.rules
#include $RULE_PATH/ddos.rules
#include $RULE_PATH/dns.rules
#include $RULE_PATH/dos.rules
#include $RULE_PATH/experimental.rules
##include $RULE_PATH/exploit-kit.rules
#include $RULE_PATH/exploit.rules
```

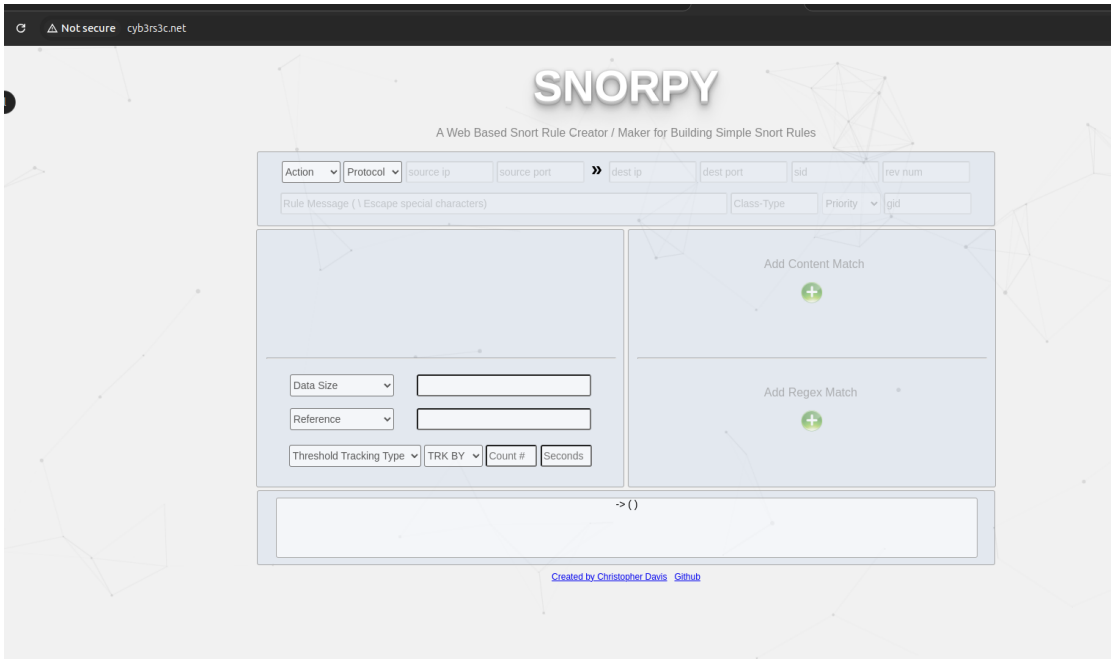
In that files I have tested 3 rules one by one.

Here I have added these 3 rules one is to detect ping and 2nd is to detect SSH authentication and 3rd is to detect nmap scan.

```
muneeb@muneeb-01:/etc/snort/rules$ cat local.rules
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# ----------------
# LOCAL RULES
# ----------------
# This file intentionally does not come with signatures.  Put your local
# additions here

alert icmp any any -> $HOME_NET any (msg:"ICMP Ping Detected"; sid:100001; rev:1;)


alert tcp any any -> $HOME_NET 22 (msg:"SSH authentication Attempt"; sid:100002; rev:01;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 1:1024 (msg:"Nmap SYN Scan detected"; flags:S; threshold:type both, track by_src, count 20, seconds 60; sid:1000005; rev:1;)
```

Unfortunately my ports were not open and I do not have second machine on my lan to test however I test ping which get logged.

Moreover I have found this very usefull online tool to create custom rules.



Step 4: Generate Network Scanning and Brute Force Traffic:

I have generated traffic using these sites.

For ping
https://dnschecker.org/ping-ipv4.php

for port scanning/nmap
https://pentest-tools.com/network-vulnerability-scanning/port-scanner-online-nmap

Step 5: Capture and Analyze the Traffic:

After that I have used the following command to sniff packets on the network interface.

sudo snort -q -l /var/log/snort -i eno1 -A fast -c /etc/snort/snort.conf

We need to run snort with elevated privlages because it is used to sniff packages and contain sensitive information.

-q options means to run snort in quite mode.

-l flag is options, means if we do not set this flag snort will log the packets in /var/log/snort anyhow, but still I explicityly mentioned that.

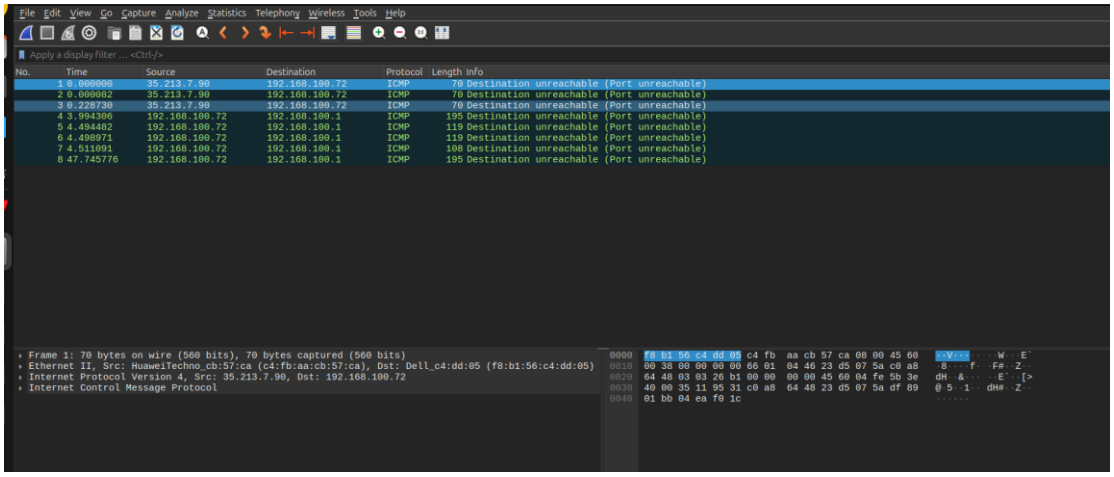
-i is used to select the network interface.

-A means to run the snort in Alert mode and from man page we can see that alert has 3 modes and from that we choose fast which is standard log form which later we can import in splunk for visualization.

-c means this is the config file that we want to run with snort usually we run main config file but we can choose different one as well and this parateter is exactly for that purpose.

Since I have choosen fast with -A flag it also logged alerts in /var/log/snort alongside packets.

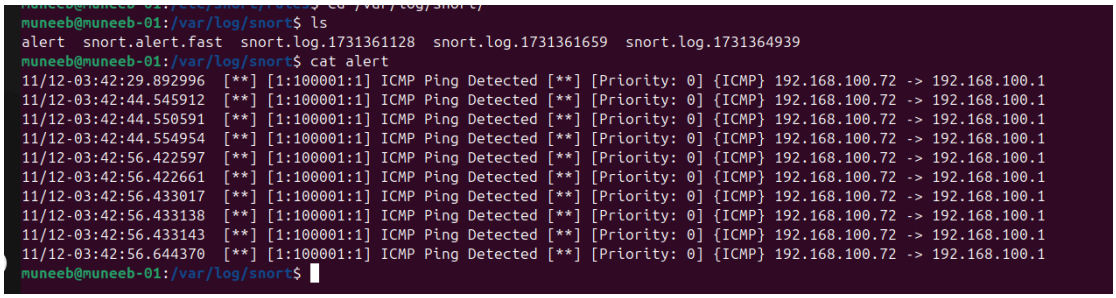
Here we can see captured packets in wireshark.



Unfortunately I only managed to get ICMP protocol which is used in ping, my ports were not open for the network so I was unable to test the other rules.

Step 6: Review Snort Alerts:

Since I used fast mode with A flag we can see alert logs here alternatively we could have used console as well to display the logs live.



Document Findings:

To conclude my experiance, snort is fairly easy yet quite powerfull tool to capture, monitor and automate network traffic. We can also set rules to drop the packet or reject the connection but those are outside the scope because they are used for Intrusion prevension system. Moreover this tool seems the definiation of customization and we can set any rules according to our needs. We can set rules for any malware detection as well and we can give content to the rule which matches the content with captured packets so the possibilites are endless.