

Objectives

In this lab students will explore the Snort Intrusion Detection Systems. The students will study Snort IDS, a signature based intrusion detection system used to detect network attacks. Snort can also be used as a simple packet logger. For the purpose of this lab the students will use snort as a packet sniffer and write their own IDS rules.

Software Requirement

All required files are packed and configured in the provided virtual machine image.

- The VMWare Software - <http://apps.eng.wayne.edu/MPStudents/Dreamspark.aspx>
- The Ubuntu 14.04 or Ubuntu Long Term Support (LTS) version or Kali linux image
- The Ubuntu 14.04 or Ubuntu 14.04 Long Term Support (LTS) Version
- Snort: A signature-based Intrusion Detection System <https://www.snort.org/#get-started>

Implementation

Starting the Lab 1 Virtual Machine

In this lab, we use Ubuntu as our VM image.

Login the Ubuntu image with username and password

Installing Snort into the Operating System

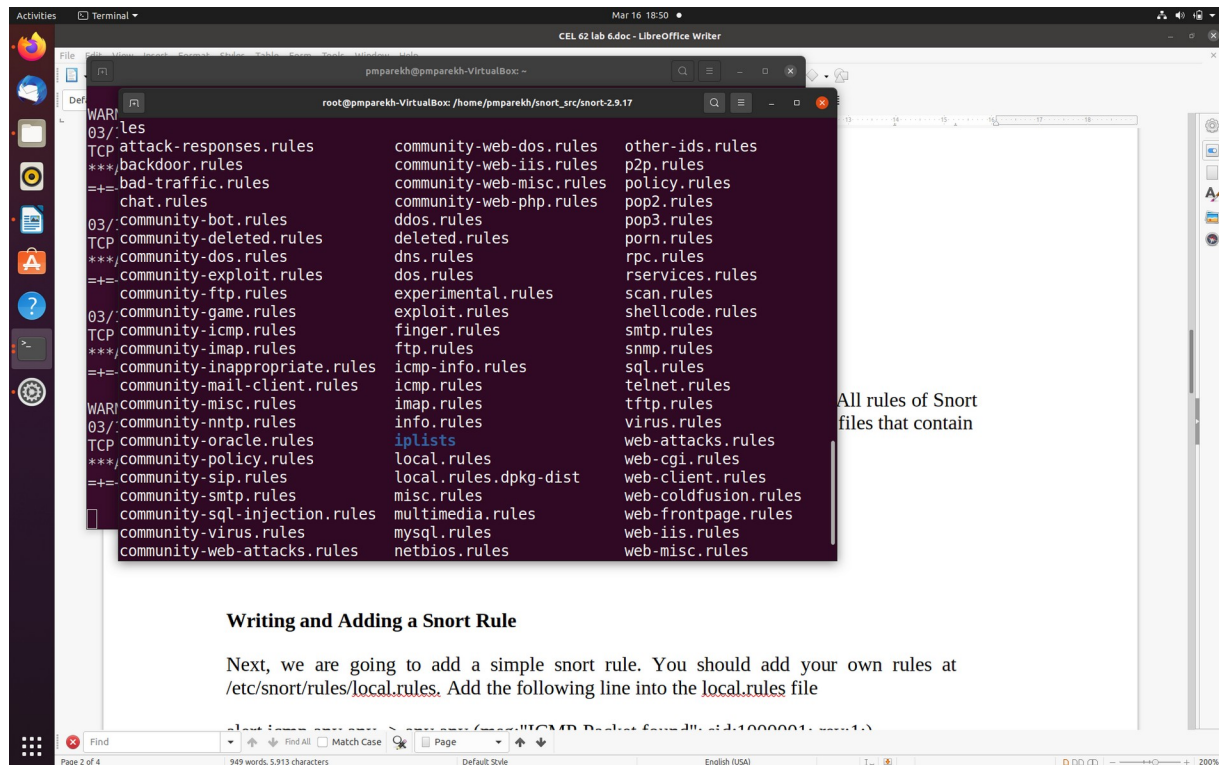
To install the latest version of the snort, you can follow the installation instruction from the snort website. Note that installation instructions are vary from OSes. The instruction below shows how to install snort from its source code on Linux.

You can find more information here:

<https://www.snort.org/#get-started>

While you install the snort, your system may miss some libraries. You need to install the required libraries, too.

Rules -



Snort is software created by Martin Roesch, which is widely used as Intrusion Prevention System [IPS] and Intrusion Detection System [IDS] in the network. It is separated into the five most important mechanisms for instance: Detection engine, Logging, and alerting system, a Packet decoder, Preprocessor, and Output modules.

The program is quite famous to carry out real-time traffic analysis, also used to detect query or attacks, packet logging on Internet Protocol networks, to detect malicious activity, denial of

service attacks and port scans by monitoring network traffic, buffer overflows, server message block probes, and stealth port scans.

Snort can be configured in three main modes:

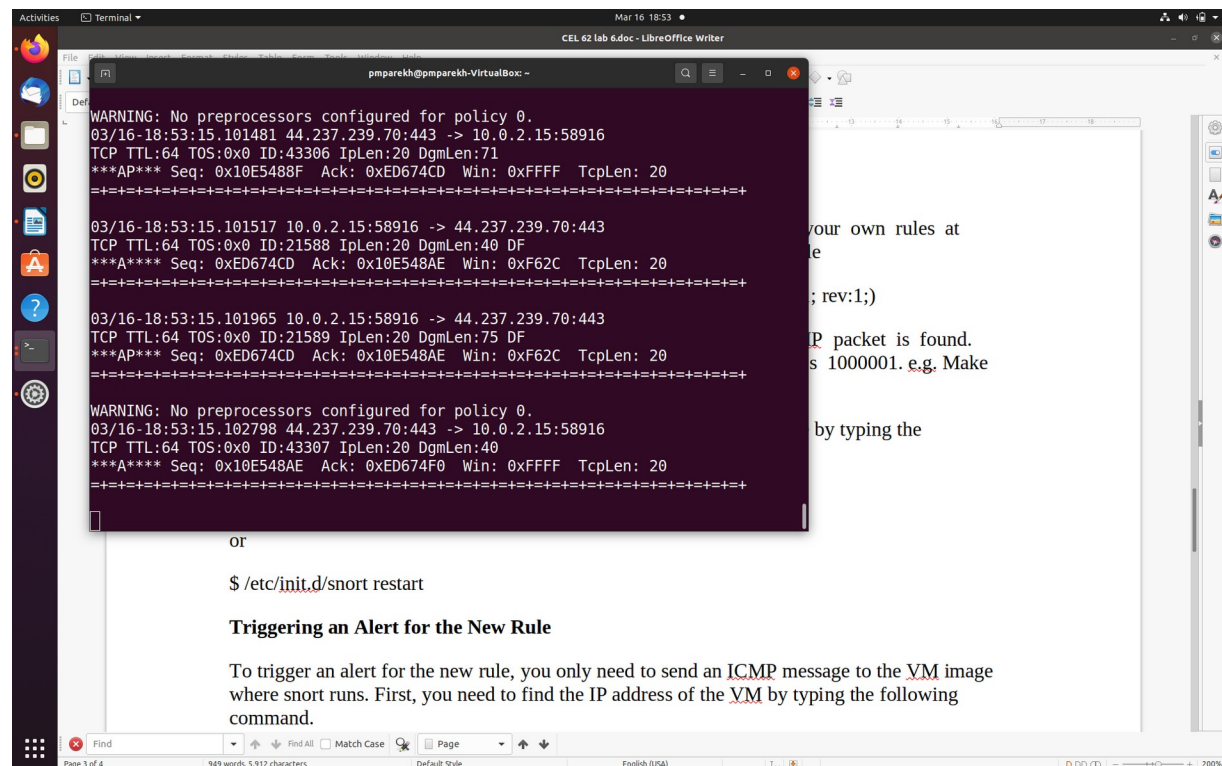
Sniffer mode: it will observe network packets and present them on the console.

Packet logger mode: it will record packets to the disk.

Intrusion detection mode: the program will monitor network traffic and analyze it against a rule set defined by the user.

After that, the application will execute a precise action depend upon what has been identified.

Starting of SNORT



Configuring and Starting the Snort IDS

After installing the Snort, we need to configure it. The configuration file of snort is stored at `/etc/snort/snort.conf`. The screenshot below shows the commands to configure the Snort. You need to switch to root to gain the permission to read the snort configurations file.

After configuring the Snort, you need to start the Snort. You can simply type the following command to start the service.

```
$ service snort start
```

or

```
$ /etc/init.d/
```

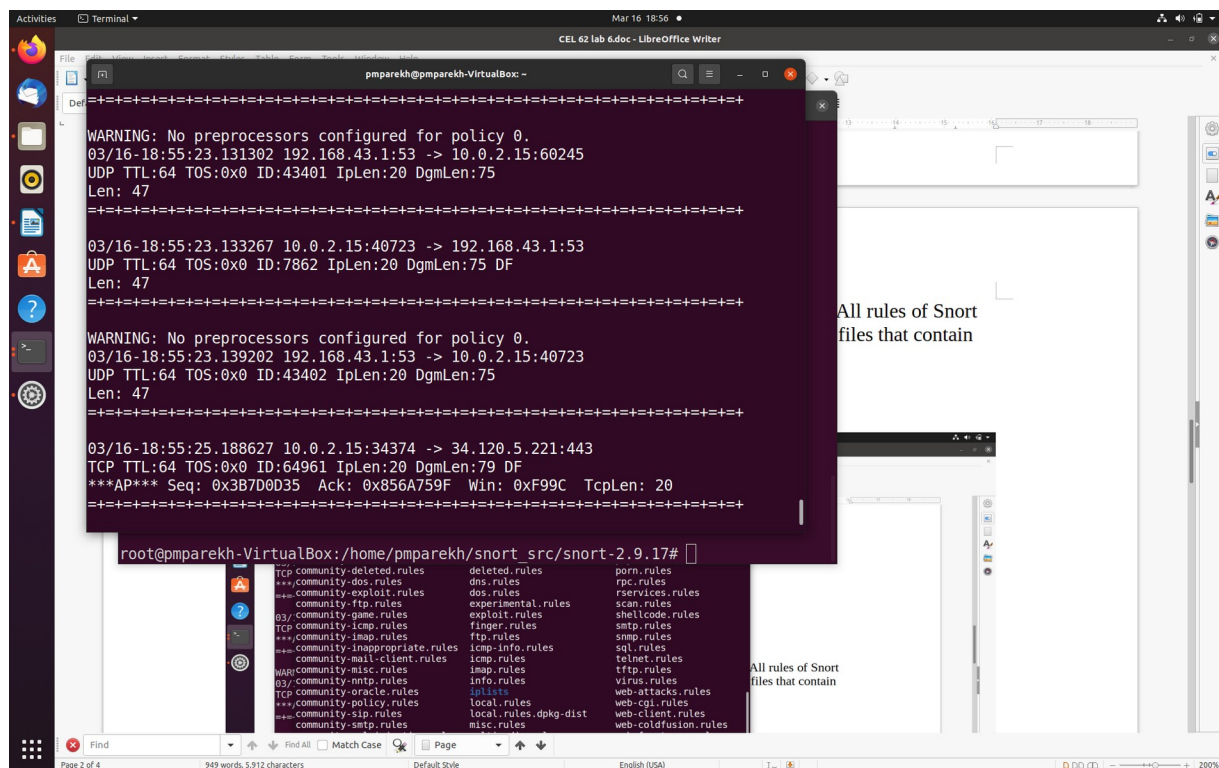
```
snort start
```

Snort Rules

Snort is a signature-based IDS, and it defines rules to detect the intrusions. All rules of Snort are stored under `/etc/snort/rules` directory. The screenshot below shows the files that contain rules of Snort.

```
$ ls /etc/snort/rules
```

Pinging with SNORT of Ubuntu from windows



Writing and Adding a Snort Rule

Next, we are going to add a simple snort rule. You should add your own rules at `/etc/snort/rules/local.rules`. Add the following line into the `local.rules` file

```
alert icmp any any -> any any (msg:"ICMP Packet found"; sid:1000001; rev:1;)
```

Basically, this rule defines that an alert will be logged if an ICMP packet is found. The ICMP packet could be from any IP address and the rule ID is 1000001. e.g. Make sure to pick a SID greater 1000000 for your own rules.

To make the rule become effective, you need to restart the snort service by typing the following command.


```

03/16-20:12:55.978264 10.0.2.15:56290 -> 192.168.43.1
TCP TTL:64 TOS:0x0 ID:54081 IPLen:20 DgmLen:84
*****S* Seq: 0xE1F0BC35 Ack: 0x0 Win: root@pmparekh-VirtualBox:/home/pmparekh/snort_src/snort-2.9.17# ls /etc/snort/rules
TCP Options (5) => MSS: 1460 S
=====
pmparekh@pmparekh-VirtualBox:~$ ping 192.168.43.1
PING 192.168.43.1 (192.168.43.1) 56(84) bytes of data.
03/16-20:12:56.011027 142.250.64 bytes from 192.168.43.1: icmp_seq=1 ttl=63 time=8.27 ms
64 bytes from 192.168.43.1: icmp_seq=2 ttl=63 time=5.82 ms
***A***S* Seq: 0x9C25A201 Ack: 64 bytes from 192.168.43.1: icmp_seq=3 ttl=63 time=10.3 ms
64 bytes from 192.168.43.1: icmp_seq=4 ttl=63 time=5.82 ms
TCP Options (1) => MSS: 1460 64 bytes from 192.168.43.1: icmp_seq=5 ttl=63 time=6.63 ms
=====
64 bytes from 192.168.43.1: icmp_seq=6 ttl=63 time=8.41 ms
03/16-20:12:56.011082 10.0.2.15:56290 -> 192.168.43.1
TCP TTL:64 TOS:0x0 ID:37467 IPLen:20 DgmLen:84
***A***S* Seq: 0x225CC212 Ack: 64 bytes from 192.168.43.1: icmp_seq=7 ttl=63 time=7.31 ms
64 bytes from 192.168.43.1: icmp_seq=8 ttl=63 time=6.52 ms
***A***S* Seq: 0x9C269C01 Ack: 64 bytes from 192.168.43.1: icmp_seq=9 ttl=63 time=4.53 ms
=====
64 bytes from 192.168.43.1: icmp_seq=10 ttl=63 time=7.97 ms
64 bytes from 192.168.43.1: icmp_seq=11 ttl=63 time=4.88 ms
WARNING: No preprocessors configured for policy 0.
03/16-20:12:56.014258 142.250.64 bytes from 192.168.43.1: icmp_seq=12 ttl=63 time=6.32 ms
64 bytes from 192.168.43.1: icmp_seq=13 ttl=63 time=6.73 ms
TCP TTL:64 TOS:0x0 ID:45354 IPLen:20 DgmLen:84
***A***S* Seq: 0x9C269C01 Ack: 64 bytes from 192.168.43.1: icmp_seq=14 ttl=63 time=5.62 ms
64 bytes from 192.168.43.1: icmp_seq=15 ttl=63 time=6.05 ms
TCP Options (1) => MSS: 1460 64 bytes from 192.168.43.1: icmp_seq=16 ttl=63 time=9.00 ms
=====
64 bytes from 192.168.43.1: icmp_seq=17 ttl=63 time=6.35 ms
64 bytes from 192.168.43.1: icmp_seq=18 ttl=63 time=6.29 ms
64 bytes from 192.168.43.1: icmp_seq=19 ttl=63 time=5.02 ms
64 bytes from 192.168.43.1: icmp_seq=20 ttl=63 time=6.53 ms
64 bytes from 192.168.43.1: icmp_seq=21 ttl=63 time=6.79 ms
64 bytes from 192.168.43.1: icmp_seq=22 ttl=63 time=3.03 ms

```

The program is quite famous to carry out real-time traffic analysis, also used to detect query or attacks, packet logging on Internet Protocol networks, to detect malicious activity, denial of service attacks and port scans by monitoring network traffic, buffer overflows, server message block probes, and stealth port scans.

\$ service snort restart

or

\$ /etc/init.d/snort restart

Pinging from Ubuntu to Windows

```

Type:8 Code:0 ID:2 Seq:11 ECHO
=====
WARNING: No preprocessors configured for policy 0.
03/16-22:08:27.675406 192.168.43.1 -> 10.0.2.15
ICMP TTL:63 TOS:0x0 ID:46780 IPLen:20 DgmLen:84
Type:0 Code:0 ID:2 Seq:11 ECHO REPLY
=====
WARNING: No preprocessors configured for policy 0.
03/16-22:08:28.670828 10.0.2.15 -> 192.168.43.1
ICMP TTL:64 TOS:0x0 ID:3076 IPLen:20 DgmLen:84 DF
Type:8 Code:0 ID:2 Seq:12 ECHO
=====
WARNING: No preprocessors configured for policy 0.
03/16-22:08:28.675071 192.168.43.1 -> 10.0.2.15
ICMP TTL:63 TOS:0x0 ID:46781 IPLen:20 DgmLen:84
Type:0 Code:0 ID:2 Seq:12 ECHO REPLY
=====
^C*** Caught Int-Signal
WARNING: No preprocessors configured for policy 0.
03/16-22:08:29.672286 10.0.2.15 -> 192.168.43.1
--- 192.168.43.1 ping statistics ---
14 packets transmitted, 14 received, 0% packet loss, time 13019ms
rtt min/avg/max/mdev = 3.475/6.888/31.737/6.939 ms
pmparekh@pmparekh-VirtualBox:~/snort_src/snort-2.9.17/etc$

```

Triggering an Alert for the New Rule

To trigger an alert for the new rule, you only need to send an ICMP message to the VM image where snort runs. First, you need to find the IP address of the VM by typing the following command.

```
$ ifconfig
```

For instance, the screenshot shows the execution result on my VM image, and the IP address is e.g. 172.16.108.242

After you have a terminal, you can just type the following command to send ping messages to the VM.

```
ping 172.16.108.242
```

After you send the ping messages, the alerts should be triggered and you can find the log messages in /var/log/snort/snort.log. However, the snort.log file will be binary format. You need to use a tool, called u2spewfoo, to read it. Observer terminal on screen with log where you can see that the SID is 1000001, and the alerts are generated by the ICMP messages.

Error no log directory

Assignments for Lab 1

1. Read the lab instructions above and finish all the tasks.

2. Answer the questions and justify your answers. Simple yes or no answer will not get any credits.

a. What is a zero-day attack? => A zero day attack means the developer doesn't have prior knowledge of the attack. A zero day is attack is very vulnerable since an attacker can exploit a vulnerability till it is found. This type of attack includes like SQL injections, buffer overflows, missing authorizations, broken algorithms.

b. Can Snort catch zero-day network attacks? If not, why not? If yes, how? => No, snort cannot catch zero-day network attacks. Since snort checks for the pre-defined rules for the zero-day attack and no developer can add all the rules for preventing the attack.

c. Given a network that has 1 million connections daily where 0.1% (not 10%) are attacks. If the IDS has a true positive rate of 95%, and the probability that an alarm is an attack is 95%. What is the false alarm rate?

Ans c. => Here, we have 1 million connections which are 10,00,000. So number of attacks are $0.1\% = 10^{-3} * 10,00,000 = \mathbf{1,000 \text{ attacks}}$. So no attacks are $99.9\% * 10,00,000 = \mathbf{9,99,000 \text{ attacks}}$. Now we have a true positive rate of 95% so the number of alarms that are set by true attacks are $95\% * 1,000 = \mathbf{950 \text{ alarms}}$. Therefore number of false alarms = $1,000 - 950 = \mathbf{50}$. Therefore the number of false alarm rate = $(\text{Number of false alarm} / \text{Total number of alarms not set}) * 100 = (50 / 9,99,000) * 100 = \mathbf{0.005\%}$.

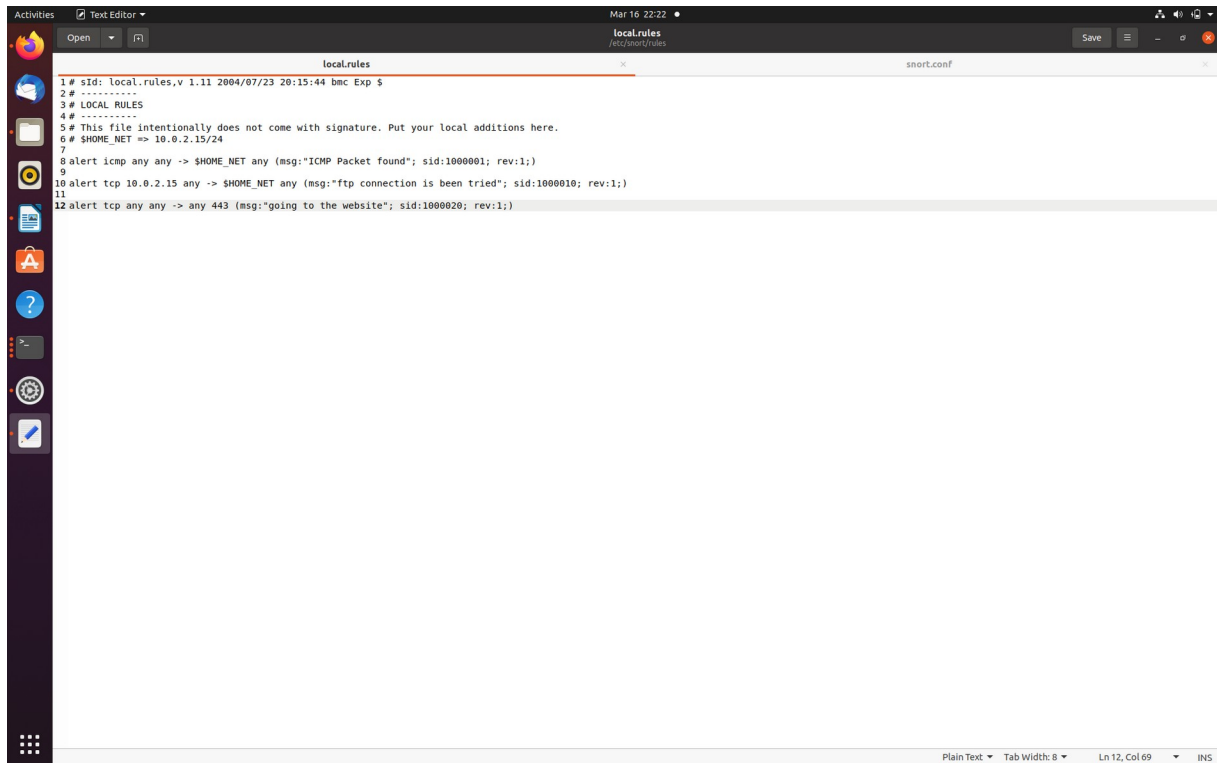
3. Write and add another snort rule and show me you trigger it.

a. The rule you added (from the rules file)

b. A description of how you triggered the alert. The alert itself from the log file (after converting it to readable text)

Extra Credit (10pt): Write a rule that will fire when you browse to any site from the machine Snort is running on; it should look for any outbound TCP request to the site you have considered and alert on it.

Rules file

A screenshot of a Linux desktop environment. The top panel shows the date and time as 'Mar 16 22:22'. The main window is a text editor titled 'local.rules' with a file path of '/etc/snort/rules'. The editor contains a Snort rules file with the following content:

```
1 # sid: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
2 # -----
3 # LOCAL RULES
4 # -----
5 # This file intentionally does not come with signature. Put your local additions here.
6 # $HOME_NET => 10.0.2.15/24
7
8 alert icmp any any -> $HOME_NET (msg:"ICMP Packet found"; sid:1000001; rev:1;)
9
10 alert tcp 10.0.2.15 any -> $HOME_NET any (msg:"ftp connection is been tried"; sid:1000010; rev:1;)
11
12 alert tcp any any -> any 443 (msg:"going to the website"; sid:1000020; rev:1;)
```

The status bar at the bottom of the text editor shows 'Plain Text', 'Tab Width: 8', 'Ln 12, Col 69', and 'INS'.

Conclusion -

After completing the above experiment, I have understood the following things -

1. A zero day attack can't be stopped by SNORT
2. SNORT requires a lot of libraries to functions. One important being libpcap. This library helps in analyzing and sniffing TCP/IP traffic.

References -

1. <https://www.youtube.com/watch?v=nSjtxIZgg8s>
2. <https://www.youtube.com/watch?v=2Yiyeu7TFbQ>