Snort IDS Configuration Lab

This document demonstrates the installation and configuration of Snort Intrusion Detection System (IDS) on Ubuntu. I describe a guide to:

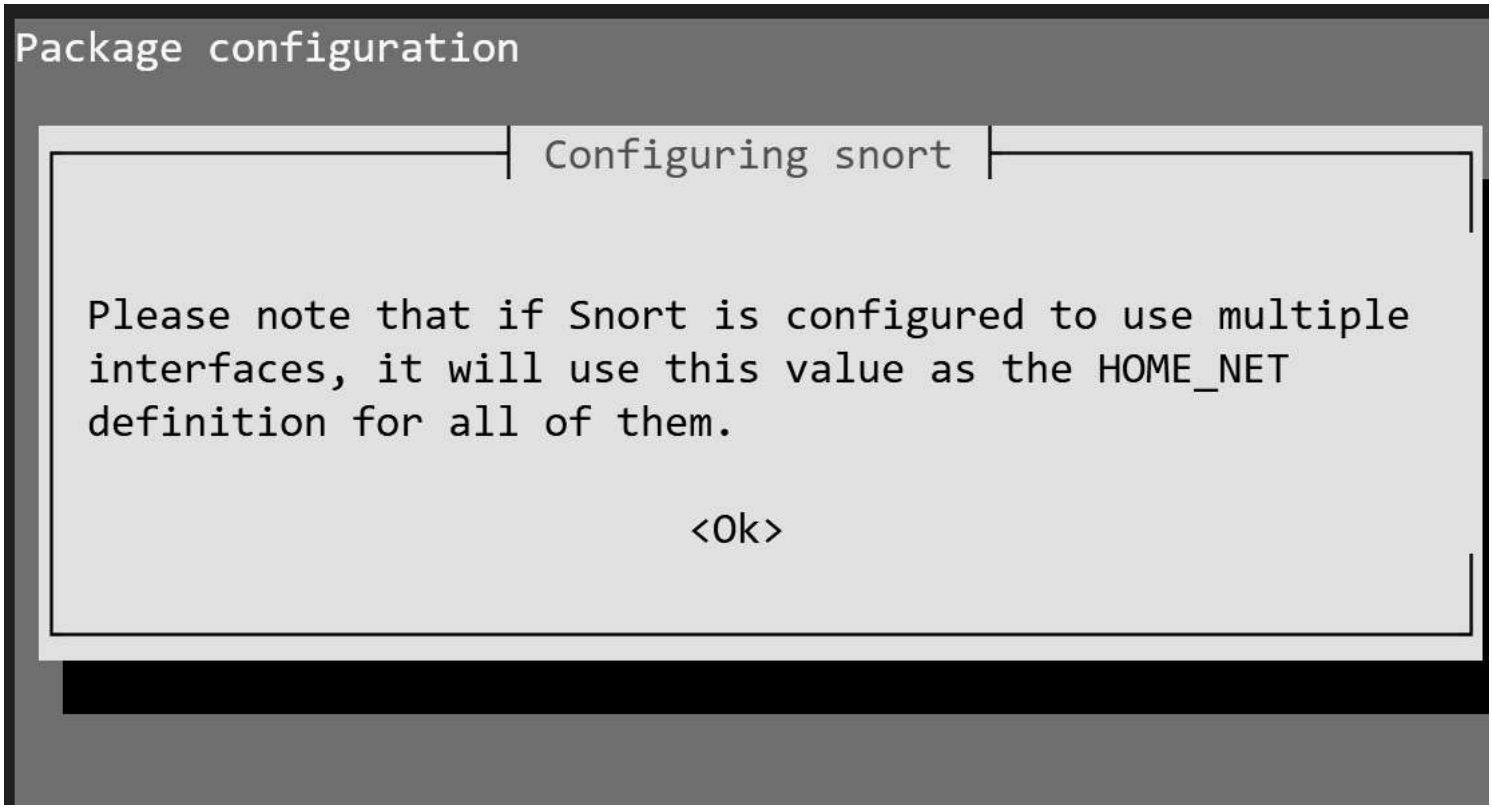
- Install and configure Snort with proper network settings
- Set up and manage detection rules for network monitoring
- Enable comprehensive logging and analyze security events
- Test and validate IDS functionality with network traffic

Step 1: Update the System

Overview:

Ensure your system is up to date before installing Snort using `sudo apt update` and `sudo apt upgrade -y`. This step guarantees compatibility and security by installing the latest packages and security patches before proceeding with Snort installation.

Step 2: Install Snort

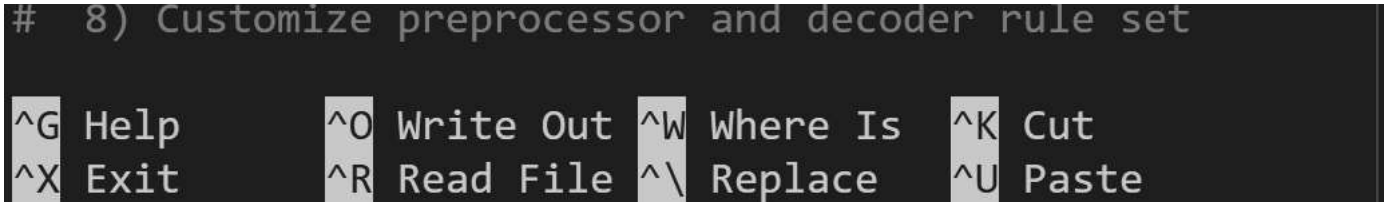


Overview:

Installing Snort directly using `sudo apt install snort -y`. During installation, you will be prompted to enter the network interface and the HOME_NET IP range that Snort will monitor. You can find your network interface using `ip a` command. Set the Network Interface (e.g., eth0, enX0, ens33) and define your home network (e.g., 192.168.1.0/24 for a private network or any to monitor all networks).

Step 3: Configure Snort

```
  GNU nano 7.2     /etc/snort/snort.conf
#--------------------------------------------------
#   VRT Rule Packages Snort.conf
#
#   For more information visit us at:
#     http://www.snort.org                    Snort We>
#     http://vrt-blog.snort.org/    Sourcefire VRT Bl>
#
#     Mailing list Contact:      snort-users@lists.sn>
#     False Positive reports:    fp@sourcefire.com
#     Snort bugs:                bugs@snort.org
#
#     Compatible with Snort Versions:
#     VERSIONS : 2.9.20
#
#     Snort build options:
#     OPTIONS : --enable-gre --enable-mpls --enable-t>
#
#     Additional information:
#     This configuration file enables active response>
#     test mode -T you are required to supply an inte>
#     or test mode will fail to fully validate the co>
#     exit with a FATAL error
#--------------------------------------------------

######################################################
# This file contains a sample snort configuration.
# You should take the following steps to create your >
#
#  1) Set the network variables.
#  2) Configure the decoder
#  3) Configure the base detection engine
#  4) Configure dynamic loaded libraries
#  5) Configure preprocessors
#  6) Configure output plugins
#  7) Customize your rule set
```



Overview:

Customize the Snort configuration by opening the main configuration file with `sudo nano /etc/snort/snort.conf`. Key sections to check include ensuring the HOME_NET variable matches your network setup. You can adjust the ipvar HOME_NET if needed (e.g., `ipvar HOME_NET 192.168.1.0/24`).

Step 4: Update and Manage Snort Rules

Overview:

By default, Snort comes with community rules, but you can download and add additional rules for better threat detection. Download community rules using:

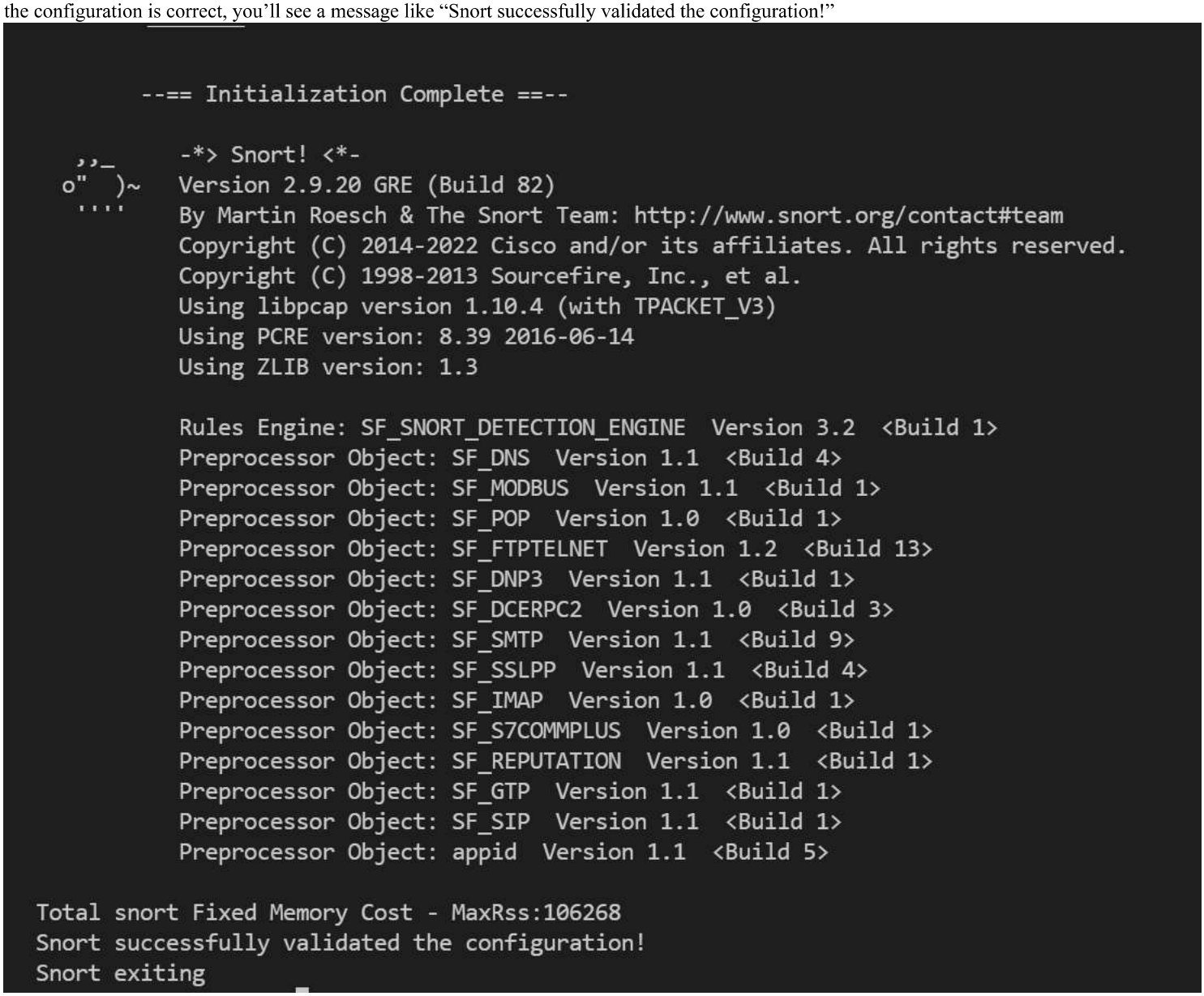
- `sudo wget https://www.snort.org/downloads/community/community-rules.tar.gz`, extract them with `sudo tar -xvzf community-rules.tar.gz`, and copy to rules directory with
- `sudo cp community-rules/* /etc/snort/rules/`.
- Add custom rules by editing `sudo nano /etc/snort/rules/local.rules`
  - Ex. `alert icmp any any -> any any (msg:"ICMP detected"; sid:1000001; rev:1;)`

```
  GNU nano 7.2      /etc/snort/rules/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
# additions here.




# Hello, and welcome:

#This rule triggers when Snort sees ICMP traffic (like ping)
alert icmp any any -> any any (msg:"ICMP detected"; sid:100000

                      [ Read 15 lines ]
^G Help     ^O Write Out^W Where Is ^K Cut      ^T Execute
^X Exit     ^R Read File^\ Replace  ^U Paste    ^J Justify
```

Step 5: Test Snort Configuration

Overview:

After configuring, test that Snort is working properly by running a configuration test with `sudo snort -T -c /etc/snort/snort.conf`. If the configuration is correct, you'll see a message like "Snort successfully validated the configuration!"



Step 6: Running Snort in IDS Mode

Overview:

Now that Snort is installed and configured, you can run it in **IDS mode** to monitor traffic by specifying the interface to monitor with

- `sudo snort -c /etc/snort/snort.conf -i eth0` (replace eth0 with your interface).

Snort will now monitor network traffic and log alerts. To exit, use Ctrl+C. This begins real-time network traffic monitoring and analysis.

```
================================================================
===============
Packet I/O Totals:
   Received:           33
   Analyzed:           30 ( 90.909%)
    Dropped:            0 (  0.000%)
   Filtered:            0 (  0.000%)
Outstanding:            3 (  9.091%)
   Injected:            0
================================================================
===============
```

```
================================================================
===============
Action Stats:
     Alerts:            4 ( 13.333%)
     Logged:            4 ( 13.333%)
     Passed:            0 (  0.000%)
Limits:
      Match:            0
      Queue:            0
        Log:            0
      Event:            0
      Alert:            0
Verdicts:
      Allow:           30 ( 90.909%)
      Block:            0 (  0.000%)
    Replace:            0 (  0.000%)
  AllowFlow:            0 (  0.000%)
  BlockFlow:            0 (  0.000%)
     Ignore:            0 (  0.000%)
      Retry:            0 (  0.000%)
================================================================
===============
```

```
                  No of frees:                 48
================================================================
===============

Memory Statistics for File at:Sun Nov  9 19:19:52 2025

Total buffers allocated:           0
Total buffers freed:               0
Total buffers released:            0
Total file mempool:                0
Total allocated file mempool:      0
Total freed file mempool:          0
Total released file mempool:       0

Heap Statistics of file:
          Total Statistics:
               Memory in use:            280 bytes
                No of allocs:              6
                 No of frees:              1
        Session Statistics:
               Memory in use:              0 bytes
                No of allocs:              1
                 No of frees:              1
        Mempool Statistics:
               Memory in use:            280 bytes
                No of allocs:              5
                 No of frees:              0
================================================================
===============
```

Step 7: Viewing Snort Logs

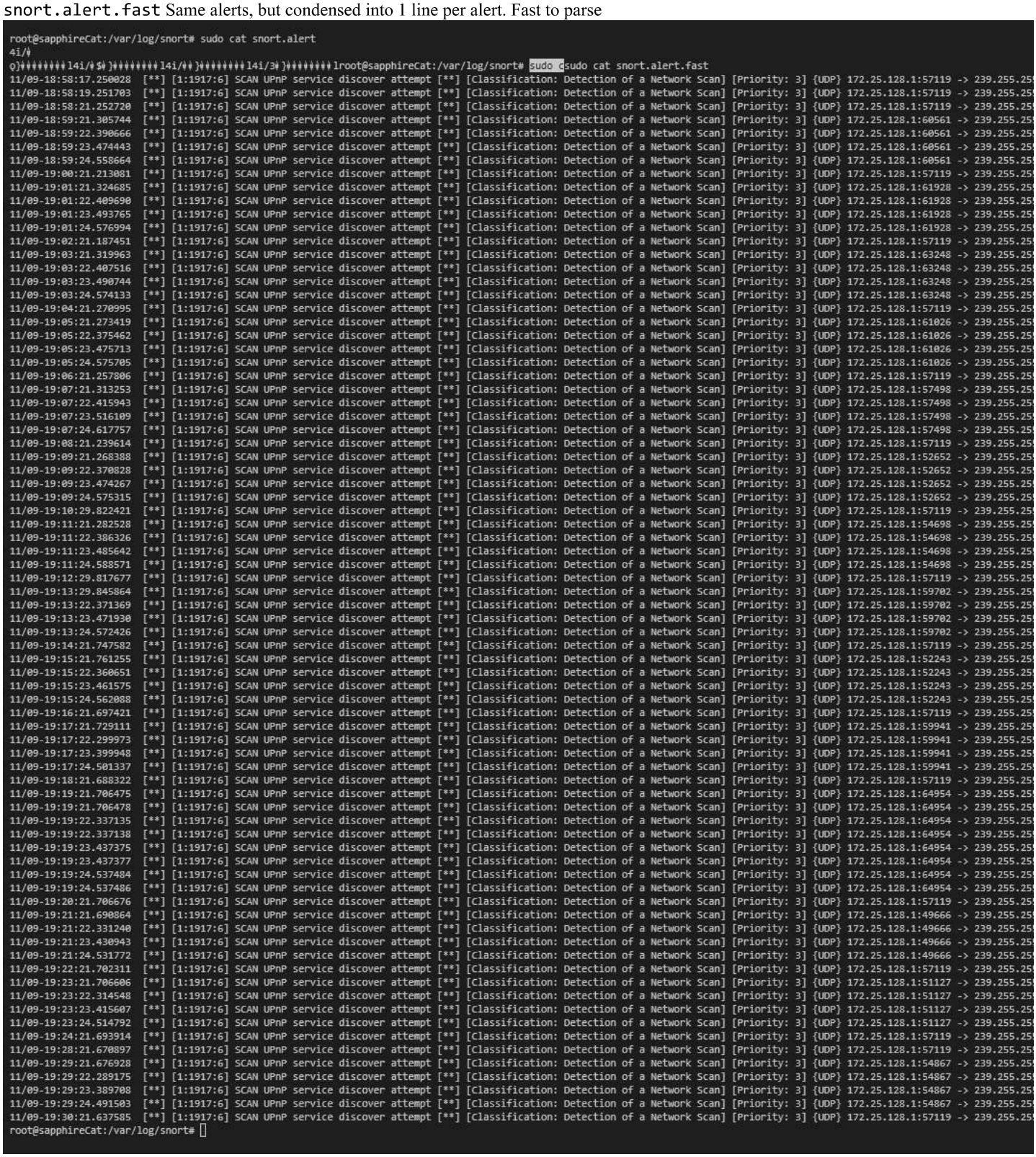
Overview:

Snort logs alerts in the `/var/log/snort/` directory. Navigate to this directory to examine what **files** are found and whether they contain any content.

Types of files:

- `snort.alert` Standard alert logging format Contains the full, detailed alerts

- `snort.alert.fast` Same alerts, but condensed into 1 line per alert. Fast to parse



- `snort.log` Raw packet capture. Not human-readable. Can be opened with Wireshark if needed.

How we can access them:

```
root@sapphireCat:~# cd /var/log/snort/
ls -l
total 24
-rw------- 1 root adm   240 Nov  9 19:19 snort.alert
-rw-r--r-- 1 root adm 14288 Nov  9 19:25 snort.alert.fast
-rw------- 1 root adm  1520 Nov  9 19:19 snort.log
root@sapphireCat:/var/log/snort#
```

Why some files were empty/garbled at the top (after catting them):

Snort only writes alerts after matching rules. Binary-looking content appears if the build writes unified2. The readable alerts are in `snort.alert.fast`

Generate some traffic so see the rule applied:

In a second terminal we can just ping a random IP:

```
root@sapphireCat:/mnt/c/Users/madel/OneDrive/Documents/School
ecurity/a6# ping -c 3 8.8.8.8
 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=112 time=31.7 ms

--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000m
root@sapphireCat:/mnt/c/Users/madel/OneDrive/Documents/School
ecurity/a6#
```

```
*] [Classification: Detection of a Network Scan] [Priority: 3
128.1:50501 -> 239.255.255.250:1900
11/09-20:17:22.712788  [**] [1:1000001:1] ICMP detected [**]
ICMP} 172.25.132.135 -> 8.8.8.8
```

```
11/09-20:15:57.046671  [**] [1:1917:6] SCAN UPnP service disc
*] [Classification: Detection of a Network Scan] [Priority: 3
```

Evidence of detections

- UPnP/SSDP background traffic repeatedly triggered rule `[1:1917:6] SCAN UPnP service discover attempt`.
- My custom rule fired: `[1:1000001:1] ICMP detected` for `172.25.132.135 <-> 8.8.8.8`.

There is a handy command which allows you to see your custom alerts live:

```
# show only your custom ICMP alerts
```

```
sudo grep 'ICMP detected' /var/log/snort/snort.alert.fast

# follow alerts live
sudo tail -f /var/log/snort/snort.alert.fast
```

and then this occurs:

```
root@sapphireCat:/var/log/snort# # show only your custom ICMP alerts
sudo grep 'ICMP detected' /var/log/snort/snort.alert.fast

# follow alerts live
sudo tail -f /var/log/snort/snort.alert.fast
11/09-20:17:22.712788  [**] [1:1000001:1] ICMP detected [**] [Priority: 0] {ICMP} 172.25.13
11/09-20:17:22.753657  [**] [1:1000001:1] ICMP detected [**] [Priority: 0] {ICMP} 8.8.8.8 -
11/09-20:17:23.812194  [**] [1:1000001:1] ICMP detected [**] [Priority: 0] {ICMP} 172.25.13
11/09-20:17:23.838766  [**] [1:1000001:1] ICMP detected [**] [Priority: 0] {ICMP} 8.8.8.8 -
11/09-20:17:24.385949  [**] [1:1000001:1] ICMP detected [**] [Priority: 0] {ICMP} 172.25.13
11/09-20:17:24.417586  [**] [1:1000001:1] ICMP detected [**] [Priority: 0] {ICMP} 8.8.8.8 -
11/09-20:15:57.076700  [**] [1:1917:6] SCAN UPnP service discover attempt [**] [Classificat
] [Priority: 3] {UDP} 172.25.128.1:50501 -> 239.255.255.250:1900
11/09-20:15:57.716176  [**] [1:1917:6] SCAN UPnP service discover attempt [**] [Classificat
] [Priority: 3] {UDP} 172.25.128.1:50501 -> 239.255.255.250:1900
11/09-20:16:01.017240  [**] [1:1917:6] SCAN UPnP service discover attempt [**] [Classificat
] [Priority: 3] {UDP} 172.25.128.1:50501 -> 239.255.255.250:1900
11/09-20:16:04.318185  [**] [1:1917:6] SCAN UPnP service discover attempt [**] [Classificat
] [Priority: 3] {UDP} 172.25.128.1:50501 -> 239.255.255.250:1900
11/09-20:17:22.712788  [**] [1:1000001:1] ICMP detected [**] [Priority: 0] {ICMP} 172.25.13
11/09-20:17:22.753657  [**] [1:1000001:1] ICMP detected [**] [Priority: 0] {ICMP} 8.8.8.8 -
11/09-20:17:23.812194  [**] [1:1000001:1] ICMP detected [**] [Priority: 0] {ICMP} 172.25.13
11/09-20:17:23.838766  [**] [1:1000001:1] ICMP detected [**] [Priority: 0] {ICMP} 8.8.8.8 -
11/09-20:17:24.385949  [**] [1:1000001:1] ICMP detected [**] [Priority: 0] {ICMP} 172.25.13
11/09-20:17:24.417586  [**] [1:1000001:1] ICMP detected [**] [Priority: 0] {ICMP} 8.8.8.8 -
```

Step 8: Running Snort as a Daemon

```
root@sapphireCat:/var/log/snort# sudo snort -D -c /etc/snort/
th0
Spawning daemon child...
My daemon child 4564 lives...
Daemon parent exiting (0)
root@sapphireCat:/var/log/snort#
```

Overview:

To run Snort in the background as a daemon, use

- `sudo snort -D -c /etc/snort/snort.conf -i eth0` (specify your interface).
- D means “daemonize”

  This will keep Snort running in the background, continuously monitoring the specified network interface. Use the `top` command to see different processes running in your system; you should see Snort running after a few seconds.

```
root@sapphireCat:/var/log/snort# ps aux | grep snort | grep -
# or
pgrep -a snort
snort       3545  0.0  1.1 136424 91968 ?        Ssl  19:31
/snort -m 027 -D -d -l /var/log/snort -u snort -g snort --pid
t/ -c /etc/snort/snort.conf -S "HOME_NET=[172.25.128.0/20]" -
root        4427  0.0  0.0  14708  5740 pts/4    S+   20:31
 -f /var/log/snort/snort.alert.fast
root        4429  0.0  0.0  14708   968 pts/3    Ss   20:31
 -f /var/log/snort/snort.alert.fast
root        4430  0.0  0.0   2728   792 pts/3    S+   20:31
var/log/snort/snort.alert.fast
root        4564  0.0  1.1 133940 88948 ?        Ssl  20:41
-c /etc/snort/snort.conf -i eth0
3545 /usr/sbin/snort -m 027 -D -d -l /var/log/snort -u snort
path /run/snort/ -c /etc/snort/snort.conf -S "HOME_NET=[172.2
 eth0
4564 snort -D -c /etc/snort/snort.conf -i eth0
root@sapphireCat:/var/log/snort#
```

here we can see 3 packets (we pinged it earlier in another terminal) being sent to my local IP address, 24.161.103.172, as an example:

```
root@sapphireCat:/var/log/snort# sudo tail -f /var/log/snort/
11/09-20:48:09.443371  [**] [1:1000001:1] ICMP detected [**]
11/09-20:48:10.028221  [**] [1:1000001:1] ICMP detected [**]
11/09-20:48:11.128539  [**] [1:1000001:1] ICMP detected [**]
11/09-20:49:05.283711  [**] [1:1000001:1] ICMP detected [**]
11/09-20:49:05.322355  [**] [1:1000001:1] ICMP detected [**]
11/09-20:49:05.876032  [**] [1:1000001:1] ICMP detected [**]
11/09-20:49:05.893819  [**] [1:1000001:1] ICMP detected [**]
11/09-20:49:06.977902  [**] [1:1000001:1] ICMP detected [**]
11/09-20:49:06.988814  [**] [1:1000001:1] ICMP detected [**]
11/09-20:49:08.111131  [**] [1:1000001:1] ICMP detected [**]
```

To stop the Snort process, you would use the terminate command to kill the process:

```
# simplest:
sudo pkill snort
```

```
# or find PID and kill
pgrep snort
sudo kill <PID>
```

Summary

This lab demonstrates Snort IDS configuration including:

- System prep + Snort installation
- Config file customization and network interface setup
- Rule management = community rules and custom local rules
- Traffic generation: Sample ping testing procedures (8.8.8.8, my local IP address)
- Real-time IDS mode for monitoring the network
- implementing a Daemon

By properly configuring rules and analyzing logs, admins can detect and respond to threats in real-time.