Snort Installation and Configuration Lab

Author: Taha Zaheer
Platform: Parrot OS (for Snort configuration) and Kali Linux (for attack simulation)
Tool Used: Snort**,** Scapy**,** Snorpy, Hping3

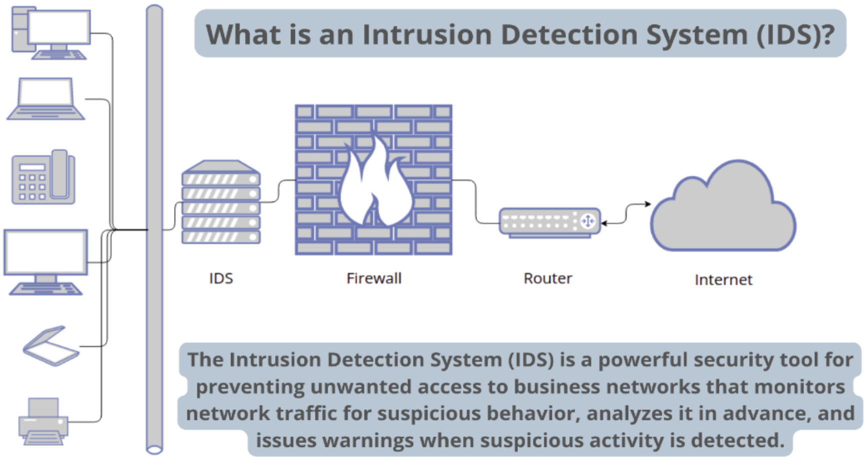
Purpose: To demonstrate the installation, configuration, and testing of **Snort** as an Intrusion Detection and Prevention System (IDS/IPS), and verify its detection capabilities using ICMP and SYN flood attacks.

Objective

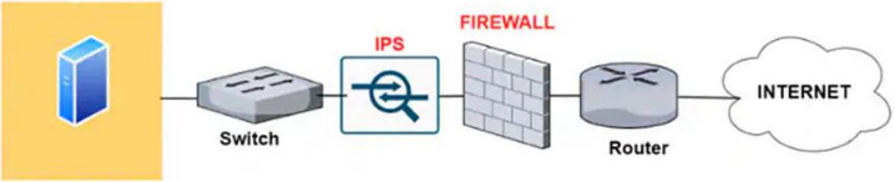
The objective of this lab is to install and configure **Snort**, an open-source Intrusion Detection System, to monitor and analyze network traffic. The lab further demonstrates how Snort can detect different types of attacks based on predefined or custom rules, such as ICMP packet monitoring and SYN flood attacks.

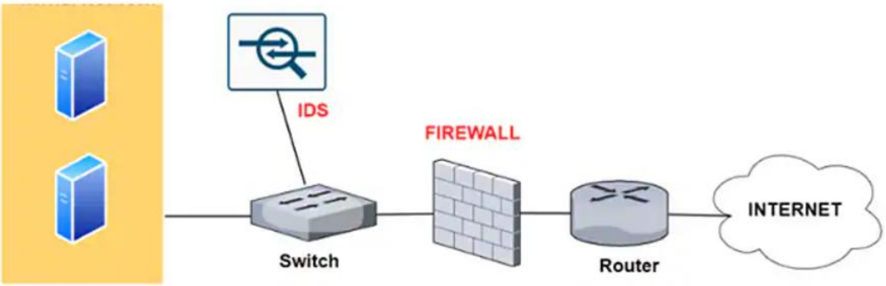
1. Understanding IDS and IPS

- **Intrusion Detection System (IDS):**
  An IDS monitors network traffic or system activities to detect unauthorized access, malicious activity, or policy violations. It operates passively by alerting the administrator upon detection of suspicious events.
- **Intrusion Prevention System (IPS):**
  An IPS is an advanced form of IDS that not only detects but also blocks or mitigates threats in real time. It operates **inline** within the network flow, actively responding to malicious packets.



The Intrusion Detection System (IDS) is a powerful security tool for preventing unwanted access to business networks that monitors network traffic for suspicious behavior, analyzes it in advance, and issues warnings when suspicious activity is detected.





Comparison Summary:

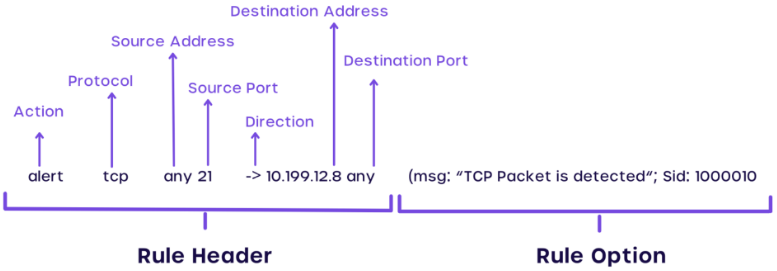
Aspect	IDS	IPS
Function	Detects and alerts	Detects and blocks
Mode	Passive	Active (Inline)
Action	No prevention	Can prevent attacks

2. Introduction to Snort

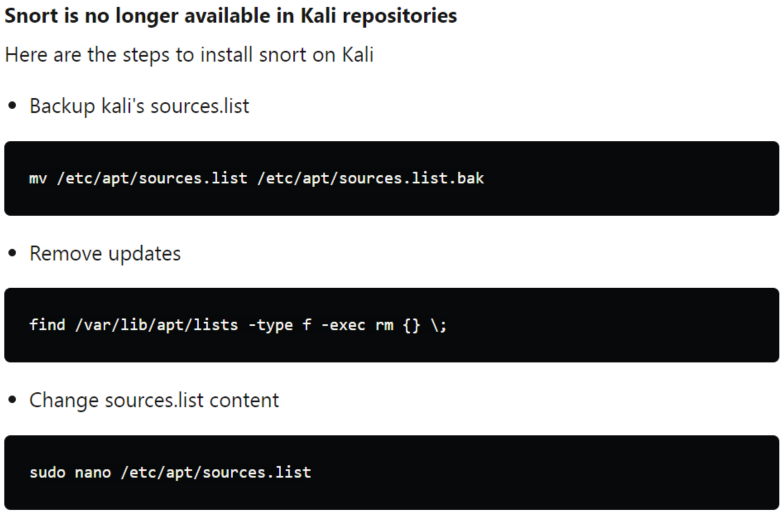
Snort is an open-source **Network Intrusion Detection and Prevention System (NIDS/NIPS)** developed by **Martin Roesch (1998)** and maintained by **Cisco**. It uses a **rule-based detection engine** to identify malicious network traffic.

Modes of Operation:

1. **Sniffer Mode:** Displays network packets on the console in real time.
2. **Packet Logger Mode:** Logs packets to disk for analysis.
3. **Network IDS/IPS Mode:** Monitors and analyzes packets based on rules to detect and prevent intrusions.



3. Snort Installation



Snort is no longer available in Kali repositories

Here are the steps to install snort on Kali

- Backup kali's sources.list

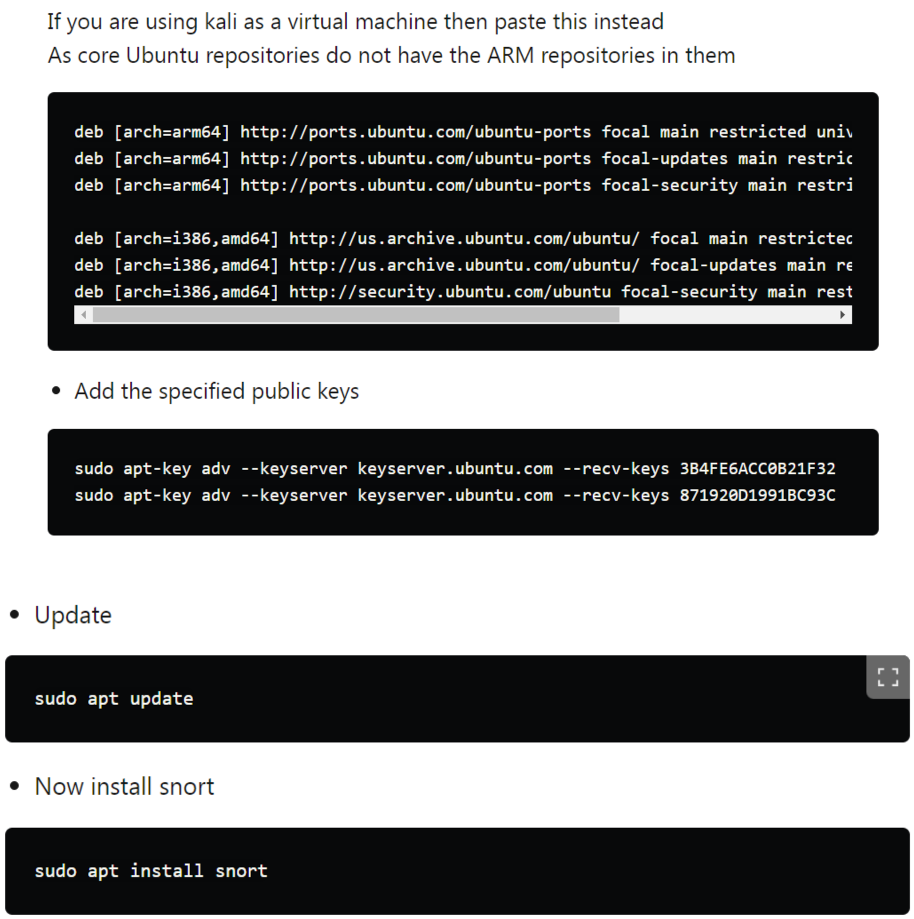
```
mv /etc/apt/sources.list /etc/apt/sources.list.bak
```

- Remove updates

```
find /var/lib/apt/lists -type f -exec rm {} \;
```

- Change sources.list content

```
sudo nano /etc/apt/sources.list
```



If you are using kali as a virtual machine then paste this instead
As core Ubuntu repositories do not have the ARM repositories in them

```
deb [arch=arm64] http://ports.ubuntu.com/ubuntu-ports focal main restricted univ
deb [arch=arm64] http://ports.ubuntu.com/ubuntu-ports focal-updates main restric
deb [arch=arm64] http://ports.ubuntu.com/ubuntu-ports focal-security main restri

deb [arch=i386,amd64] http://us.archive.ubuntu.com/ubuntu/ focal main restrictec
deb [arch=i386,amd64] http://us.archive.ubuntu.com/ubuntu/ focal-updates main re
deb [arch=i386,amd64] http://security.ubuntu.com/ubuntu focal-security main rest
```

- Add the specified public keys

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 3B4FE6ACC0B21F32
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 871920D1991BC93C
```

- Update

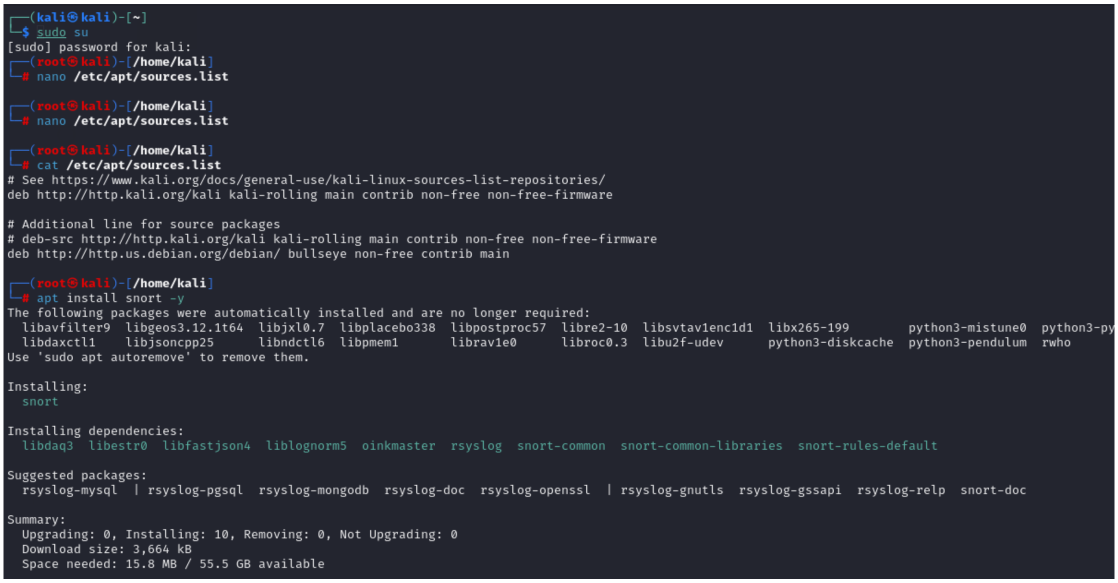
```
sudo apt update
```

- Now install snort

```
sudo apt install snort
```

Verify installation:

```
snort --version
```



```
┌──(kali㉿kali)-[~]
└─$ sudo su
[sudo] password for kali:
┌──(root㉿kali)-[/home/kali]
└─# nano /etc/apt/sources.list

┌──(root㉿kali)-[/home/kali]
└─# nano /etc/apt/sources.list

┌──(root㉿kali)-[/home/kali]
└─# cat /etc/apt/sources.list
# See https://www.kali.org/docs/general-use/kali-linux-sources-list-repositories/
deb http://http.kali.org/kali kali-rolling main contrib non-free non-free-firmware

# Additional line for source packages
# deb-src http://http.kali.org/kali kali-rolling main contrib non-free non-free-firmware
deb http://http.us.debian.org/debian/ bullseye non-free contrib main

┌──(root㉿kali)-[/home/kali]
└─# apt install snort -y
The following packages were automatically installed and are no longer required:
  libavfilter9  libgeos3.12.1t64  libjxl0.7  libplacebo338  libpostproc57  libre2-10  libsvtav1enc1d1  libx265-199       python3-mistune0  python3-py
  libdaxctl1    libjsoncpp25      libndctl6  libpmem1       librav1e0      libroc0.3  libu2f-udev      python3-diskcache  python3-pendulum  rwho
Use 'sudo apt autoremove' to remove them.

Installing:
  snort

Installing dependencies:
  libdaq3  libestr0  libfastjson4  liblognorm5  oinkmaster  rsyslog  snort-common  snort-common-libraries  snort-rules-default

Suggested packages:
  rsyslog-mysql  | rsyslog-pgsql  rsyslog-mongodb  rsyslog-doc  rsyslog-openssl  | rsyslog-gnutls  rsyslog-gssapi  rsyslog-relp  snort-doc

Summary:
  Upgrading: 0, Installing: 10, Removing: 0, Not Upgrading: 0
  Download size: 3,664 kB
  Space needed: 15.8 MB / 55.5 GB available
```

4. Configuring Snort

Step 2: Edit the Configuration File

Open the Snort configuration file:

```
sudo nano /etc/snort/snort.conf
```

Set the home network variable:

```
ipvar HOME_NET 150.1.7.0/24
```

Define log directory:

```
config logdir: /var/log/snort
```

5. Creating and Managing Snort Rules

Navigate to the rules directory:

```
cd /etc/snort/rules
```

Create a new ICMP rule file:

```
sudo nano icmp.rules
```

Add a custom detection rule for ICMP packets:

```
alert icmp any any -> 150.1.7.0/24 (msg:"ICMP Packet Found"; sid:10000001;)
```

Save and exit.

6. Running Snort

Run Snort in console mode to monitor ICMP packets in real time:

```
sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i ens34
```

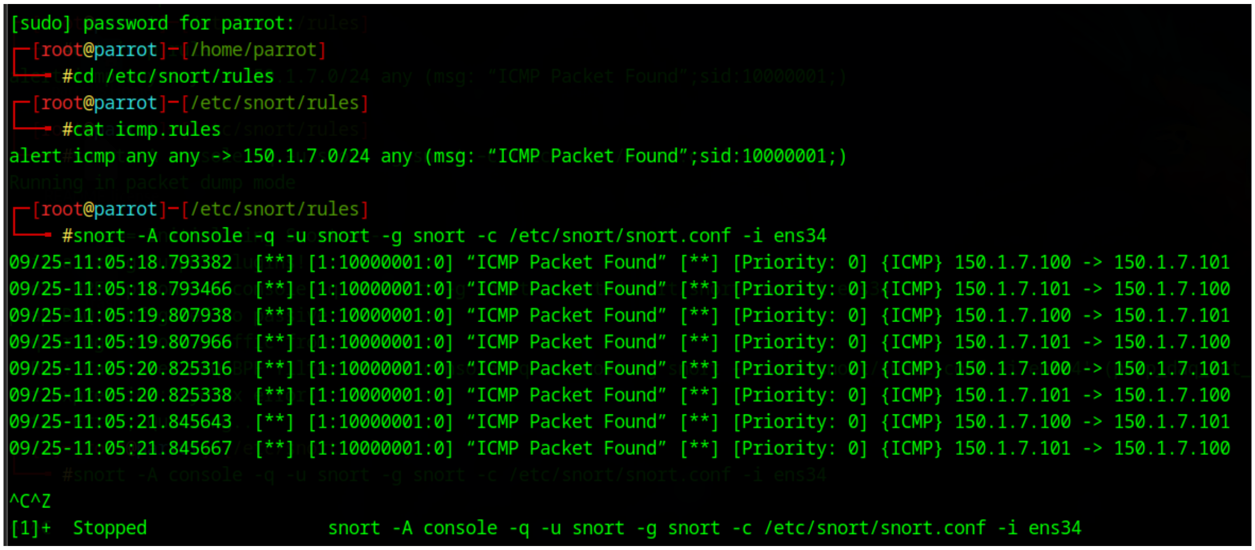
From another machine, send ICMP requests:

```
ping -n 2 150.1.7.104
```

7. Testing Snort with Attack Simulations

A. ICMP Attack Detection

Snort successfully detected ICMP packets sent to the configured HOME_NET and displayed alerts in the console with the message “ICMP Packet Found.”



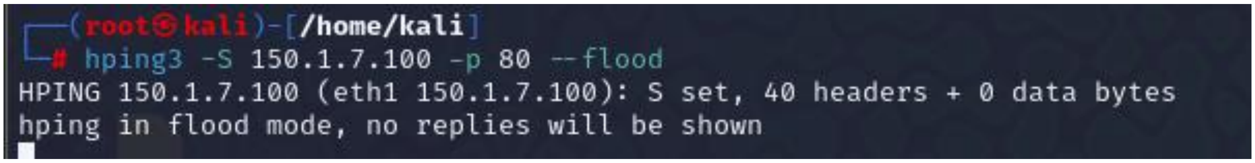
```
[sudo] password for parrot:
┌─[root@parrot]─[/home/parrot]
└──╼ #cd /etc/snort/rules
┌─[root@parrot]─[/etc/snort/rules]
└──╼ #cat icmp.rules
alert icmp any any -> 150.1.7.0/24 any (msg: “ICMP Packet Found”;sid:10000001;)
┌─[root@parrot]─[/etc/snort/rules]
└──╼ #snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i ens34
09/25-11:05:18.793382  [**] [1:10000001:0] “ICMP Packet Found” [**] [Priority: 0] {ICMP} 150.1.7.100 -> 150.1.7.101
09/25-11:05:18.793466  [**] [1:10000001:0] “ICMP Packet Found” [**] [Priority: 0] {ICMP} 150.1.7.101 -> 150.1.7.100
09/25-11:05:19.807938  [**] [1:10000001:0] “ICMP Packet Found” [**] [Priority: 0] {ICMP} 150.1.7.100 -> 150.1.7.101
09/25-11:05:19.807966  [**] [1:10000001:0] “ICMP Packet Found” [**] [Priority: 0] {ICMP} 150.1.7.101 -> 150.1.7.100
09/25-11:05:20.825316  [**] [1:10000001:0] “ICMP Packet Found” [**] [Priority: 0] {ICMP} 150.1.7.100 -> 150.1.7.101
09/25-11:05:20.825338  [**] [1:10000001:0] “ICMP Packet Found” [**] [Priority: 0] {ICMP} 150.1.7.101 -> 150.1.7.100
09/25-11:05:21.845643  [**] [1:10000001:0] “ICMP Packet Found” [**] [Priority: 0] {ICMP} 150.1.7.100 -> 150.1.7.101
09/25-11:05:21.845667  [**] [1:10000001:0] “ICMP Packet Found” [**] [Priority: 0] {ICMP} 150.1.7.101 -> 150.1.7.100
^C^Z
[1]+  Stopped                 snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i ens34
```

B. SYN Flood Attack Using hping3

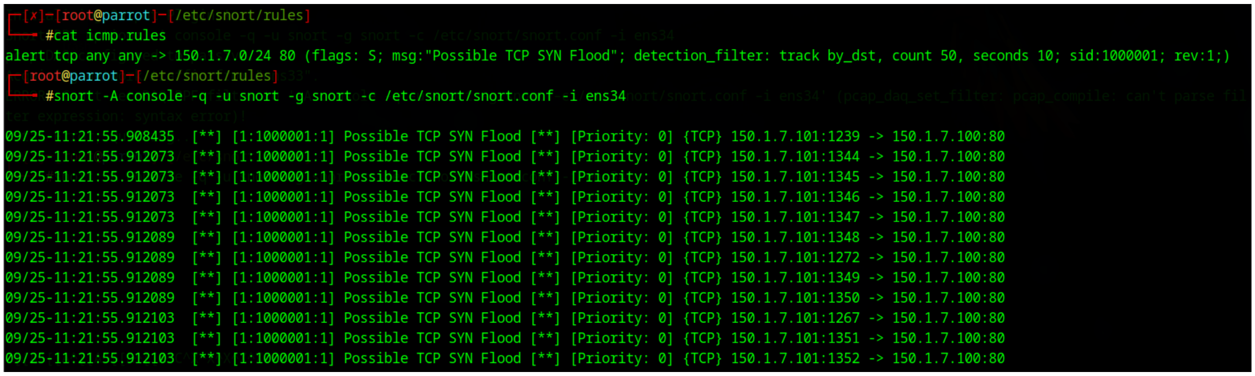
Simulate a SYN Flood attack using the `hping3` tool from Kali Linux:

```
sudo hping3 -S --flood -V -p 22 150.1.7.104
```

Snort detects the flood of SYN packets based on its internal or user-defined rules and raises alerts for potential denial-of-service activity.



```
┌──(root㉿kali)-[/home/kali]
└─# hping3 -S 150.1.7.100 -p 80 --flood
HPING 150.1.7.100 (eth1 150.1.7.100): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

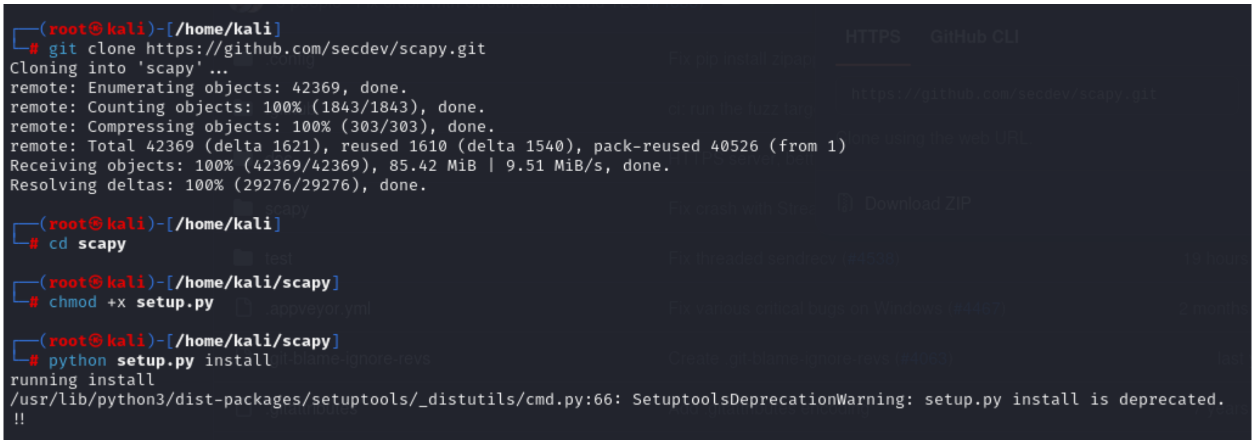


```
┌─[✗]─[root@parrot]─[/etc/snort/rules]
└──╼ #cat icmp.rules
alert tcp any any -> 150.1.7.0/24 80 (flags: S; msg:"Possible TCP SYN Flood"; detection_filter: track by_dst, count 50, seconds 10; sid:1000001; rev:1;)
┌─[root@parrot]─[/etc/snort/rules]
└──╼ #snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i ens34
09/25-11:21:55.908435  [**] [1:1000001:1] Possible TCP SYN Flood [**] [Priority: 0] {TCP} 150.1.7.101:1239 -> 150.1.7.100:80
09/25-11:21:55.912073  [**] [1:1000001:1] Possible TCP SYN Flood [**] [Priority: 0] {TCP} 150.1.7.101:1344 -> 150.1.7.100:80
09/25-11:21:55.912073  [**] [1:1000001:1] Possible TCP SYN Flood [**] [Priority: 0] {TCP} 150.1.7.101:1345 -> 150.1.7.100:80
09/25-11:21:55.912073  [**] [1:1000001:1] Possible TCP SYN Flood [**] [Priority: 0] {TCP} 150.1.7.101:1346 -> 150.1.7.100:80
09/25-11:21:55.912073  [**] [1:1000001:1] Possible TCP SYN Flood [**] [Priority: 0] {TCP} 150.1.7.101:1347 -> 150.1.7.100:80
09/25-11:21:55.912089  [**] [1:1000001:1] Possible TCP SYN Flood [**] [Priority: 0] {TCP} 150.1.7.101:1348 -> 150.1.7.100:80
09/25-11:21:55.912089  [**] [1:1000001:1] Possible TCP SYN Flood [**] [Priority: 0] {TCP} 150.1.7.101:1272 -> 150.1.7.100:80
09/25-11:21:55.912089  [**] [1:1000001:1] Possible TCP SYN Flood [**] [Priority: 0] {TCP} 150.1.7.101:1349 -> 150.1.7.100:80
09/25-11:21:55.912089  [**] [1:1000001:1] Possible TCP SYN Flood [**] [Priority: 0] {TCP} 150.1.7.101:1350 -> 150.1.7.100:80
09/25-11:21:55.912103  [**] [1:1000001:1] Possible TCP SYN Flood [**] [Priority: 0] {TCP} 150.1.7.101:1267 -> 150.1.7.100:80
09/25-11:21:55.912103  [**] [1:1000001:1] Possible TCP SYN Flood [**] [Priority: 0] {TCP} 150.1.7.101:1351 -> 150.1.7.100:80
09/25-11:21:55.912103  [**] [1:1000001:1] Possible TCP SYN Flood [**] [Priority: 0] {TCP} 150.1.7.101:1352 -> 150.1.7.100:80
```

C. SYN Flood Attack Using Scapy

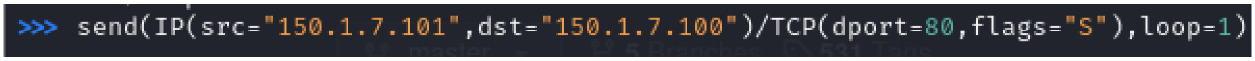
Install Scapy for packet crafting:

```
sudo apt install python3-scapy
```

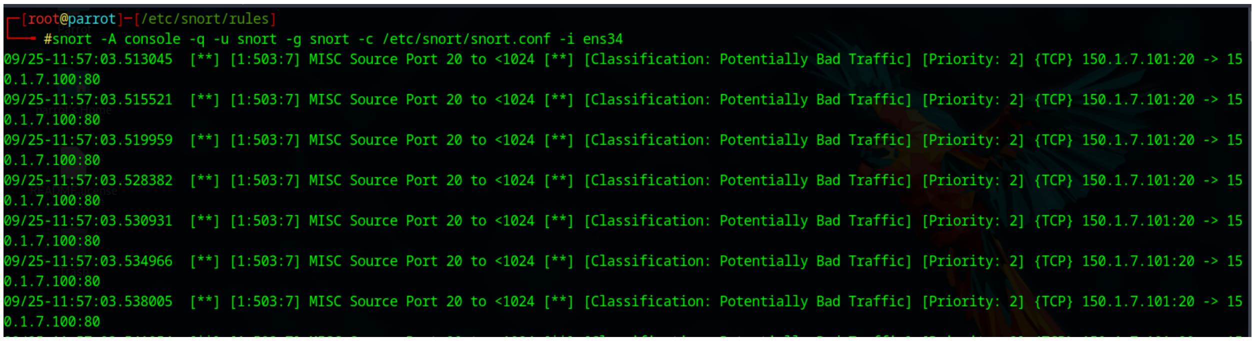


```
┌──(root㉿kali)-[/home/kali]
└─# git clone https://github.com/secdev/scapy.git
Cloning into 'scapy'...
remote: Enumerating objects: 42369, done.
remote: Counting objects: 100% (1843/1843), done.
remote: Compressing objects: 100% (303/303), done.
remote: Total 42369 (delta 1621), reused 1610 (delta 1540), pack-reused 40526 (from 1)
Receiving objects: 100% (42369/42369), 85.42 MiB | 9.51 MiB/s, done.
Resolving deltas: 100% (29276/29276), done.

┌──(root㉿kali)-[/home/kali]
└─# cd scapy

┌──(root㉿kali)-[/home/kali/scapy]
└─# chmod +x setup.py

┌──(root㉿kali)-[/home/kali/scapy]
└─# python setup.py install
running install
/usr/lib/python3/dist-packages/setuptools/_distutils/cmd.py:66: SetuptoolsDeprecationWarning: setup.py install is deprecated.
!!
```

Then execute a SYN flood attack script (example from Scapy documentation) to verify Snort's response.



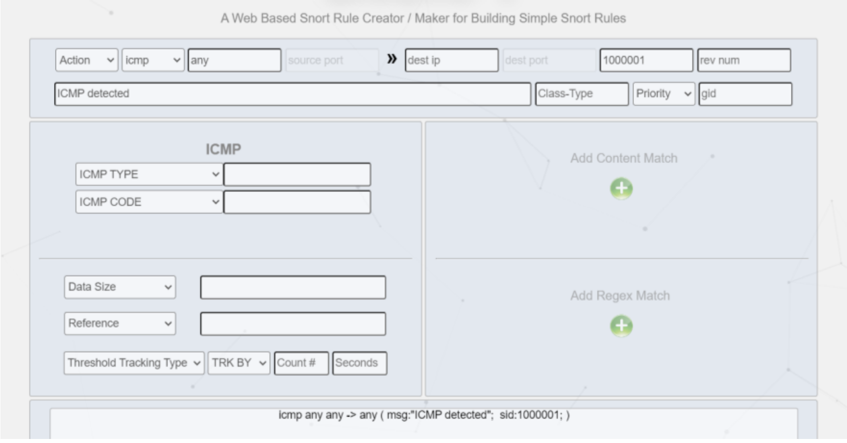
```
>>> send(IP(src="150.1.7.101",dst="150.1.7.100")/TCP(dport=80,flags="S"),loop=1)
```



```
┌[root@parrot]─[/etc/snort/rules]
└──╼ #snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i ens34
09/25-11:57:03.513045  [**] [1:503:7] MISC Source Port 20 to <1024 [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 150.1.7.101:20 -> 15
0.1.7.100:80
09/25-11:57:03.515521  [**] [1:503:7] MISC Source Port 20 to <1024 [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 150.1.7.101:20 -> 15
0.1.7.100:80
09/25-11:57:03.519959  [**] [1:503:7] MISC Source Port 20 to <1024 [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 150.1.7.101:20 -> 15
0.1.7.100:80
09/25-11:57:03.528382  [**] [1:503:7] MISC Source Port 20 to <1024 [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 150.1.7.101:20 -> 15
0.1.7.100:80
09/25-11:57:03.530931  [**] [1:503:7] MISC Source Port 20 to <1024 [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 150.1.7.101:20 -> 15
0.1.7.100:80
09/25-11:57:03.534966  [**] [1:503:7] MISC Source Port 20 to <1024 [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 150.1.7.101:20 -> 15
0.1.7.100:80
09/25-11:57:03.538005  [**] [1:503:7] MISC Source Port 20 to <1024 [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 150.1.7.101:20 -> 15
0.1.7.100:80
```

8. Using Snorpy for Rule Generation

To simplify custom rule creation, **Snorpy**, a web-based Snort rule generator, can be used. It provides a GUI interface for defining conditions and outputs a properly formatted Snort rule.



```
icmp any any -> any ( msg:"ICMP detected";  sid:1000001; )
```

Conclusion

This lab demonstrates the **complete installation, configuration, and testing of Snort IDS/IPS**. Snort was able to detect ICMP and SYN flood attacks successfully, validating its capability as a robust, open-source intrusion detection solution.