IPS_using_SNORT tool

IPS (Intrusion Prevention System)

An **Intrusion Prevention System (IPS)** is a security technology designed to monitor network or system activities and automatically take actions to block or prevent identified threats in real-time. While an **Intrusion Detection System (IDS)** simply detects potential intrusions and alerts security personnel, an IPS actively prevents or mitigates attacks, making it a more proactive defence mechanism.

Key Functions of IPS

1. **Detection**: Like an IDS, an IPS monitors network or system traffic for malicious activity or security breaches, identifying threats through various detection methods.
2. **Prevention**: The key difference between an IDS and IPS is that an IPS can take action to stop or block malicious traffic. This could involve dropping suspicious packets, blocking IP addresses, or terminating connections.
3. **Alerting**: IPS systems can also alert security personnel about detected and blocked threats for further investigation.
4. **Logging**: An IPS logs the actions taken (e.g., blocking traffic) and may include additional data such as source/destination IP addresses, attack type, and severity of the event.

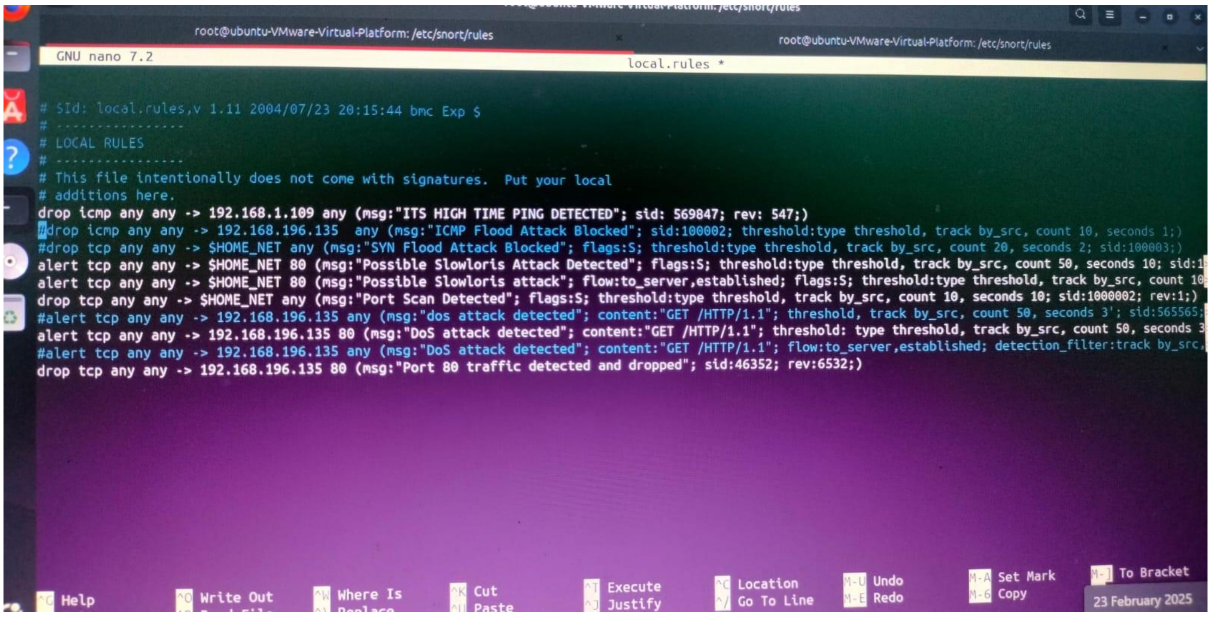
Proof of concept:

Step-1:

On the ubuntu terminal: open the rules as **cd/etc/snort/rules**

In **local.rules:**

drop imp any any -> 192.168.1.109 any (msg:"ITS HIGH TIME PING DETECTED"; sid: 569847; rev: 547;)



```
  GNU nano 7.2                                              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.
drop icmp any any -> 192.168.1.109 any (msg:"ITS HIGH TIME PING DETECTED"; sid: 569847; rev: 547;)
#drop icmp any any -> 192.168.196.135  any (msg:"ICMP Flood Attack Blocked"; sid:100002; threshold:type threshold, track by_src, count 10, seconds 1;)
#drop tcp any any -> $HOME_NET any (msg:"SYN Flood Attack Blocked"; flags:S; threshold:type threshold, track by_src, count 20, seconds 2; sid:100003;)
alert tcp any any -> $HOME_NET 80 (msg:"Possible Slowloris Attack Detected"; flags:S; threshold:type threshold, track by_src, count 50, seconds 10; sid:1
alert tcp any any -> $HOME_NET 80 (msg:"Possible Slowloris attack"; flow:to_server,established; flags:S; threshold:type threshold, track by_src, count 10
drop tcp any any -> $HOME_NET any (msg:"Port Scan Detected"; flags:S; threshold:type threshold, track by_src, count 10, seconds 10; sid:1000002; rev:1;)
#alert tcp any any -> 192.168.196.135 any (msg:"dos attack detected"; content:"GET /HTTP/1.1"; threshold, track by_src, count 50, seconds 3'; sid:565565;
alert tcp any any -> 192.168.196.135 80 (msg:"DoS attack detected"; content:"GET /HTTP/1.1"; threshold: type threshold, track by_src, count 50, seconds 3
#alert tcp any any -> 192.168.196.135 any (msg:"DoS attack detected"; content:"GET /HTTP/1.1"; flow:to_server,established; detection_filter:track by_src,
drop tcp any any -> 192.168.196.135 80 (msg:"Port 80 traffic detected and dropped"; sid:46352; rev:6532;)
```

step-2:

iptables -A INPUT -p icmp -j NFQUEUE --queue -num 0

The command you provided is an iptables rule used to interact with Netfilter's NFQUEUE to direct network traffic to user-space for processing. Let's break it down to understand each part:

iptables -A INPUT -p icmp -j NFQUEUE --queue-num 0

1. iptables

- iptables is a command-line utility for configuring and managing the packet filtering rules of the Linux kernel's Netfilter subsystem. It allows you to control incoming and outgoing network traffic by specifying rules for filtering, logging, and modifying packets.

2. -A INPUT

- -A stands for Append. This means the rule is being added to the INPUT chain of the firewall configuration.
- The INPUT chain handles packets destined for the local system.
- So, this rule applies to packets coming into the machine (i.e., traffic directed to the local system).

3. -p icmp

- -p specifies the protocol for the rule.

- In this case, icmp refers to Internet Control Message Protocol. ICMP is used for network diagnostics (e.g., ping command) and error reporting in networks.

4. -j NFQUEUE

- -j stands for jump. It specifies the target action to take when the packet matches the rule.
- NFQUEUE is an action that sends the matching packet to a specific Netfilter queue for processing by user-space applications.
- The packet will not be dropped, allowed, or rejected immediately; instead, it is handed off to a user-space application (e.g., a program that processes the packet).

5. --queue-num 0

- --queue-num specifies the queue number for the packet to be sent to.
- 0 means the packet will be sent to queue 0. In practice, this queue is often managed by an application running in user-space that is configured to process the packets in queue 0.
- You can have multiple queues with different numbers (e.g., --queue-num 1, --queue-num 2) to organize traffic for different purposes or different applications.

What does this rule do?

This rule tells iptables to:

- Capture incoming ICMP traffic (e.g., ping requests or other ICMP-based communications) using the -p icmp option.
- Send it to NFQUEUE (instead of allowing or dropping the packet immediately).
- Place it in queue 0 for processing by a user-space application (which would need to be configured to read and process packets from queue 0).

Step-3:

snort -Q --daq nfq --daq-var queue=0 -c /etc/snort/snort.conf -A console

1. snort

- This is the executable for Snort, which starts the IDS/IPS process. Running this command invokes Snort and begins network traffic analysis.

2. -Q

- The **-Q** option tells Snort to **run in **"NFQ" mode**, which means it will interact with the **Netfilter Queue (NFQUEUE)** for packet capture.
- This option enables Snort to read and process packets that have been queued by **iptables** using the NFQUEUE target, which is commonly used to redirect traffic to user-space programs like Snort for further inspection.

3. --daq nfq

- **--daq** stands for **Data Acquisition (DAQ)** module. DAQ modules allow Snort to interface with different packet capture mechanisms.
- **nfq** refers to **Netfilter Queue**, which tells Snort to use the Netfilter framework (via the NFQUEUE target) to capture packets.
- This option essentially allows Snort to get the packets from the Netfilter Queue (set up earlier by iptables with the rule -j NFQUEUE --queue-num 0), where they can be processed by Snort in user-space.

4. --daq-var queue=0

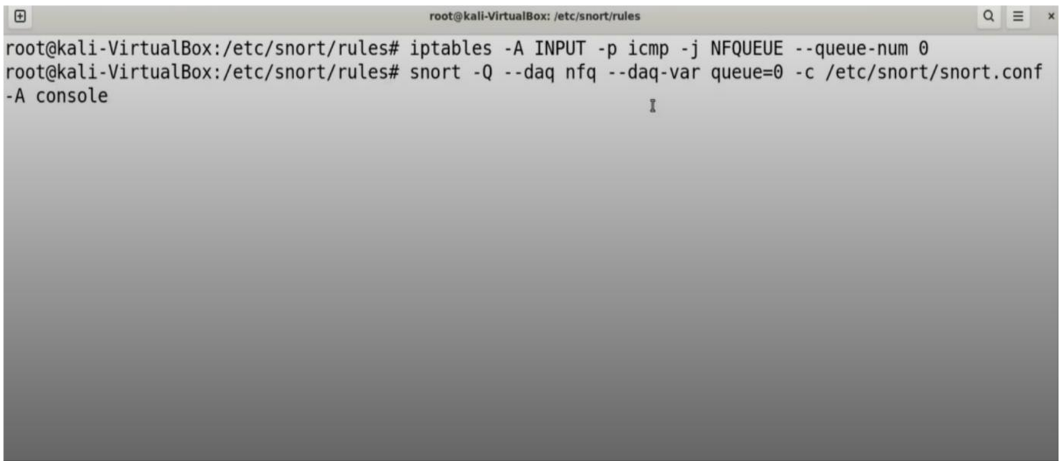
- **--daq-var** specifies additional variables for the DAQ module.
- **queue=0** tells Snort to use **queue 0** from Netfilter, which was configured by the earlier iptables rule. This ensures that Snort processes the packets from the same queue where they were directed by iptables using the **NFQUEUE** target.

5. -c /etc/snort/snort.conf

- The **-c** option specifies the path to the **Snort configuration file**.
- In this case, the file being referenced is **/etc/snort/snort.conf**, which contains Snort's rules, settings, and configuration options that define how Snort should analyse network traffic.
- This file contains the rules for detecting various types of attacks and vulnerabilities.

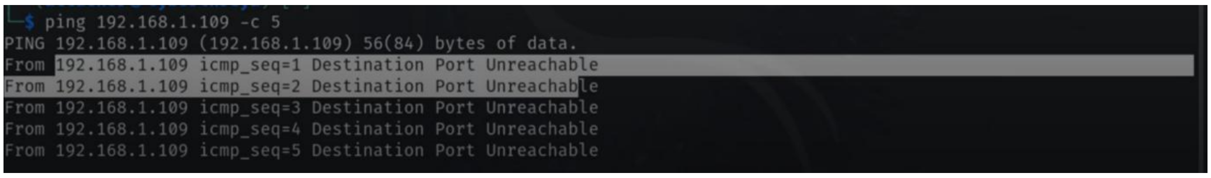
6. -A console

- The **-A** option specifies how Snort should output alerts.
- **console** means that Snort will display alerts directly on the **console** (standard output). This is useful for testing or when you want to monitor alerts in real-time as they are generated by Snort.
- Other output methods (e.g., to files or databases) can be used as alternatives, but **console** is typically used for immediate feedback during testing or debugging.



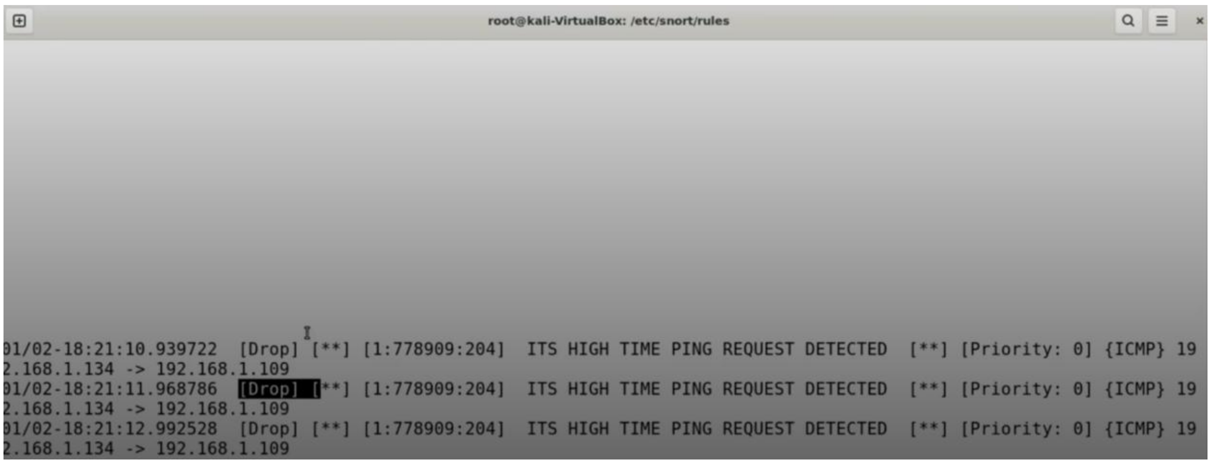
```
root@kali-VirtualBox:/etc/snort/rules# iptables -A INPUT -p icmp -j NFQUEUE --queue-num 0
root@kali-VirtualBox:/etc/snort/rules# snort -Q --daq nfq --daq-var queue=0 -c /etc/snort/snort.conf
-A console
```

The target machine is kali Linux ,here we are performing the PING scan



```
└─$ ping 192.168.1.109 -c 5
PING 192.168.1.109 (192.168.1.109) 56(84) bytes of data.
From 192.168.1.109 icmp_seq=1 Destination Port Unreachable
From 192.168.1.109 icmp_seq=2 Destination Port Unreachable
From 192.168.1.109 icmp_seq=3 Destination Port Unreachable
From 192.168.1.109 icmp_seq=4 Destination Port Unreachable
From 192.168.1.109 icmp_seq=5 Destination Port Unreachable
```

So,here we observed that Destination port unreachable message that implies snort has blocked the ip



```
01/02-18:21:10.939722  [Drop] [**] [1:778909:204]  ITS HIGH TIME PING REQUEST DETECTED  [**] [Priority: 0] {ICMP} 19
2.168.1.134 -> 192.168.1.109
01/02-18:21:11.968786  [Drop] [**] [1:778909:204]  ITS HIGH TIME PING REQUEST DETECTED  [**] [Priority: 0] {ICMP} 19
2.168.1.134 -> 192.168.1.109
01/02-18:21:12.992528  [Drop] [**] [1:778909:204]  ITS HIGH TIME PING REQUEST DETECTED  [**] [Priority: 0] {ICMP} 19
2.168.1.134 -> 192.168.1.109
```

Finally the ALERT message has appeared.

If running the command “snort -Q --daq nfq --daq-var queue=0 -c /etc/snort/snort.conf -A console” if any errors occurs can be also try this

“PS aux | grep snort

kill -p <new line>”.