Name Hamza Ali
Code Alpa Task 2

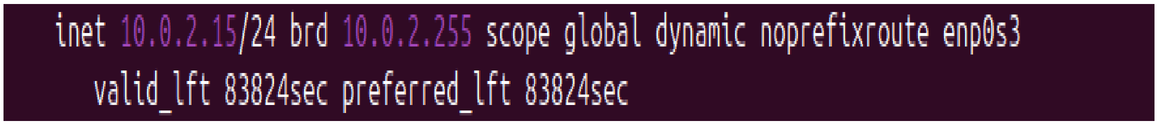
Network Intrusion Detection System (NIDS) Report

1. Introduction

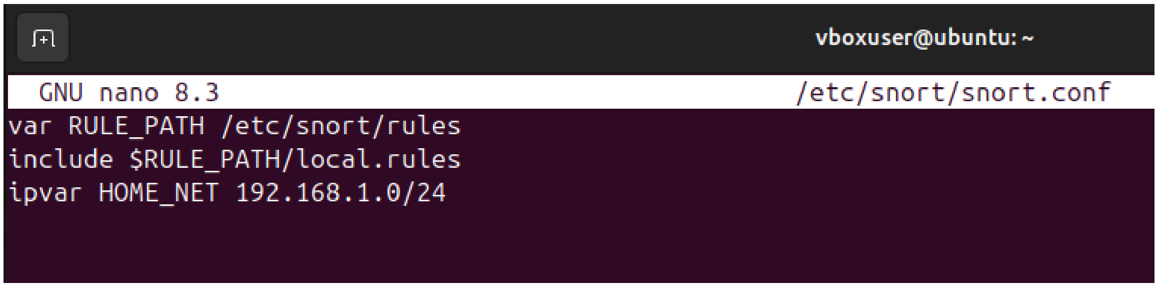
This report documents the setup and configuration of a Network-based Intrusion Detection System (NIDS) using Snort on a Kali Linux environment. It includes configuration of custom alert rules, generating test alerts, and monitoring for suspicious activity.

2. Environment Setup

- OS: Ubuntu
- Tool: Snort
- Interface Monitored: enp0s3 (or equivalent active interface)



```
   inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
      valid_lft 83824sec preferred_lft 83824sec
```



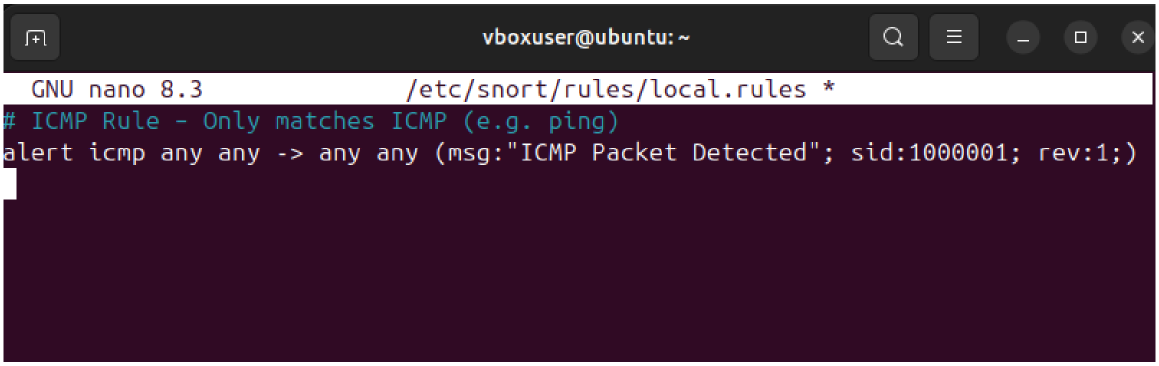
```
  GNU nano 8.3                                       /etc/snort/snort.conf
var RULE_PATH /etc/snort/rules
include $RULE_PATH/local.rules
ipvar HOME_NET 192.168.1.0/24
```

3. Rule Configuration

A custom rule was created to detect ICMP echo requests (ping scans):

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

This rule was added to the local.rules file and included in snort.conf.

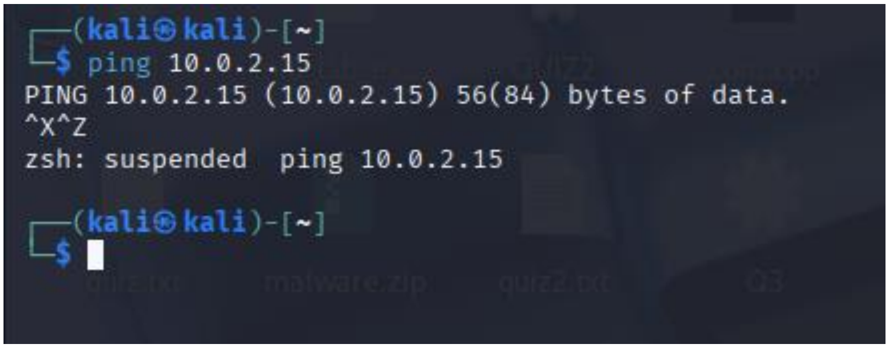


```
  GNU nano 8.3              /etc/snort/rules/local.rules *
# ICMP Rule - Only matches ICMP (e.g. ping)
alert icmp any any -> any any (msg:"ICMP Packet Detected"; sid:1000001; rev:1;)
```

4. Generating Alerts

To generate alerts, the following steps were taken:

- Another machine was used to send ICMP packets using the ping command.



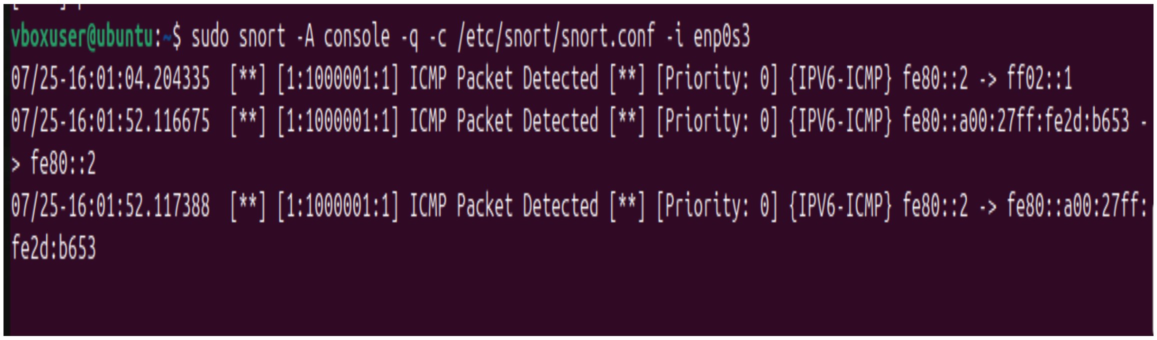
```
┌──(kali㉿kali)-[~]
└─$ ping 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.
^X^Z
zsh: suspended  ping 10.0.2.15

┌──(kali㉿kali)-[~]
└─$ 
```

- The Snort service was running in packet sniffing mode:

sudo snort -A console -q -c /etc/snort/snort.conf -i eth0

An alert was successfully triggered and displayed in the console.



```
vboxuser@ubuntu:~$ sudo snort -A console -q -c /etc/snort/snort.conf -i enp0s3
07/25-16:01:04.204335  [**] [1:1000001:1] ICMP Packet Detected [**] [Priority: 0] {IPV6-ICMP} fe80::2 -> ff02::1
07/25-16:01:52.116675  [**] [1:1000001:1] ICMP Packet Detected [**] [Priority: 0] {IPV6-ICMP} fe80::a00:27ff:fe2d:b653 -
> fe80::2
07/25-16:01:52.117388  [**] [1:1000001:1] ICMP Packet Detected [**] [Priority: 0] {IPV6-ICMP} fe80::2 -> fe80::a00:27ff:
fe2d:b653
```

5. Results and Response

Snort successfully detected the ICMP traffic and displayed the alert message. In a real-world scenario, an automatic response mechanism (such as blocking IPs using iptables or fail2ban) can be configured.

6. Conclusion

The NIDS setup using Snort successfully detected and alerted on suspicious ICMP traffic. This demonstrates how custom rules and basic monitoring can help identify network threats in real-time.