

code_alpha-task2-setting-of-network-intrusion-detection-system

Network Intrusion Detection System (NIDS) using Snort, implemented on Ubuntu with Kali Linux attack simulations. Detects real-time threats like port scans and exploits through customized rules. Validated in a virtual lab environment, demonstrating effective traffic monitoring and alerting. Provides a foundation for network security hardening

Objectives

Deploy Snort on Ubuntu as a NIDS to monitor a designated network.

Simulate reconnaissance attacks from Kali Linux using:

Nmap (port scanning, OS fingerprinting)

Evaluate Snort's detection accuracy by analyzing generated alerts.

Document rules and configurations for reproducibility.

Part I: Lab Environment Setup

1. Virtual Infrastructure Attacker Machine: Kali Linux (Offensive Security Tools)

Monitored Machine: Ubuntu 22.04 LTS (Running Snort IDS)

Network: Host-only/NAT configuration to isolate lab traffic

```
root@LAPTOP-K2DSLUE: ~  
root@LAPTOP-K2DSLUE:~# ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 172.24.195.150 netmask 255.255.240.0 broadcast 172.24.207.255  
    inet6 fe80::215:5dff:fea5:b9e7 prefixlen 64 scopeid 0x20<link>  
    ether 00:15:5d:a5:b9:e7 txqueuelen 1000 (Ethernet)  
    RX packets 17 bytes 4244 (4.2 KB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 14 bytes 1032 (1.0 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 8 bytes 1631 (1.6 KB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 8 bytes 1631 (1.6 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
root@LAPTOP-K2DSLUE:~# |
```

Figure 1.1: IP address and Network Interface Card for Ubuntu

```
akanksha@LAPTOP-K2DSLUE: ~  
(Message from Kali developers)  
This is a minimal installation of Kali Linux, you likely  
want to install supplementary tools. Learn how:  
=> https://www.kali.org/docs/troubleshooting/common-minimum-setup/  
  
(Run: "touch ~/.hushlogin" to hide this message)  
(akanksha@LAPTOP-K2DSLUE)~  
$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 172.24.195.150 netmask 255.255.240.0 broadcast 172.24.207.255  
    inet6 fe80::215:5dff:fea5:b9e7 prefixlen 64 scopeid 0x20<link>  
    ether 00:15:5d:a5:b9:e7 txqueuelen 1000 (Ethernet)  
    RX packets 22 bytes 4550 (4.4 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 22 bytes 1548 (1.5 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 8 bytes 1631 (1.5 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 8 bytes 1631 (1.5 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
(akanksha@LAPTOP-K2DSLUE)~  
$
```

Figure 1.2: Verifying attacking machine IP address

```
root@LAPTOP-K2DSLUEU: ~  
---- Initialization Complete ----  
  
--> Snort! <--  
Version 2.9.20 GRE (Build 82)  
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team  
Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.  
Copyright (C) 1998-2013 Sourcefire, Inc., et al.  
Using libpcap version 1.10.4 (with TPACKET_V3)  
Using PCRE version: 8.39 2016-06-14  
Using ZLIB version: 1.3  
  
Rules Engine: SF_SNORT_DETECTION_ENGINE Version 3.2 <Build 1>  
Preprocessor Object: SF_SMTP Version 1.1 <Build 9>  
Preprocessor Object: SF_FTPTELNET Version 1.2 <Build 13>  
Preprocessor Object: SF_REPUTATION Version 1.1 <Build 1>  
Preprocessor Object: SF_GTP Version 1.1 <Build 1>  
Preprocessor Object: SF_SDF Version 1.1 <Build 1>  
Preprocessor Object: appid Version 1.1 <Build 5>  
Preprocessor Object: SF_DCEP2C2 Version 1.0 <Build 3>  
Preprocessor Object: SF_SSH Version 1.1 <Build 3>  
Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>  
Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>  
Preprocessor Object: SF_MODBUS Version 1.1 <Build 1>  
Preprocessor Object: SF_DNS Version 1.1 <Build 4>  
Preprocessor Object: SF_S7COMPLUS Version 1.0 <Build 1>  
Preprocessor Object: SF_IMAP Version 1.0 <Build 1>  
Preprocessor Object: SF_SIP Version 1.1 <Build 1>  
Preprocessor Object: SF_POP Version 1.0 <Build 1>  
  
Total snort Fixed Memory Cost - MaxRss:106004  
Snort successfully validated the configuration!  
Snort exiting  
root@LAPTOP-K2DSLUEU:~#
```

Figure 1.3: Snort successfully validated the configuration

Part II: Attack Simulation & Detection Analysis

Simulated Attacks from Kali

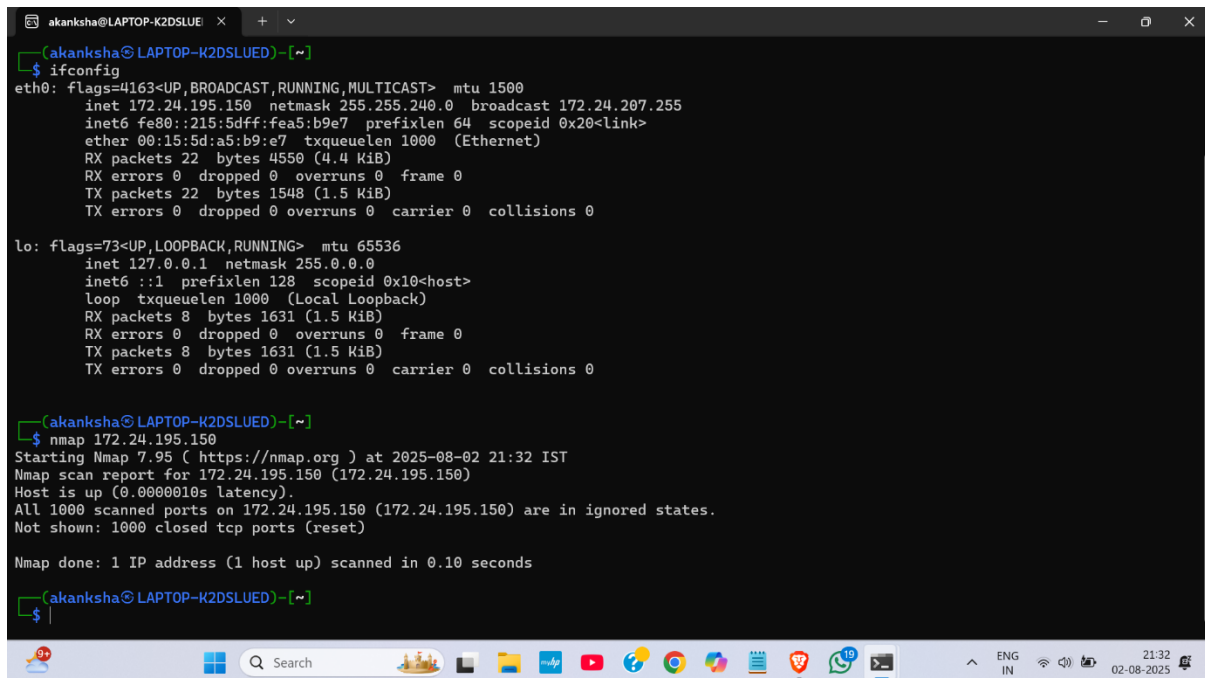
Linux

```
root@LAPTOP-K2DSLUEU: ~  
Using ZLIB version: 1.3  
  
Rules Engine: SF_SNORT_DETECTION_ENGINE Version 3.2 <Build 1>  
Preprocessor Object: SF_SMTP Version 1.1 <Build 9>  
Preprocessor Object: SF_FTPTELNET Version 1.2 <Build 13>  
Preprocessor Object: SF_REPUTATION Version 1.1 <Build 1>  
Preprocessor Object: SF_GTP Version 1.1 <Build 1>  
Preprocessor Object: SF_SDF Version 1.1 <Build 1>  
Preprocessor Object: appid Version 1.1 <Build 5>  
Preprocessor Object: SF_DCEP2C2 Version 1.0 <Build 3>  
Preprocessor Object: SF_SSH Version 1.1 <Build 3>  
Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>  
Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>  
Preprocessor Object: SF_MODBUS Version 1.1 <Build 1>  
Preprocessor Object: SF_DNS Version 1.1 <Build 4>  
Preprocessor Object: SF_S7COMPLUS Version 1.0 <Build 1>  
Preprocessor Object: SF_IMAP Version 1.0 <Build 1>  
Preprocessor Object: SF_SIP Version 1.1 <Build 1>  
Preprocessor Object: SF_POP Version 1.0 <Build 1>  
  
Total snort Fixed Memory Cost - MaxRss:106004  
Snort successfully validated the configuration!  
Snort exiting  
root@LAPTOP-K2DSLUEU:~# sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
```

Figure 2.1: Monitoring enabled

Kali Linux initiated an Nmap scan against an Ubuntu machine to test Snort's reconnaissance detection. The scan probed for open ports,

services, and OS fingerprints, generating detectable network traffic. Snort (if properly configured) should log alerts for scan patterns (e.g., SYN floods, service probes). This simulation validates Snort's ability to detect basic Nmap scans as part of intrusion detection.



```
(akanksha@LAPTOP-K2DSLUE)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.24.195.150 netmask 255.255.240.0 broadcast 172.24.207.255
    inet6 fe80::215:5dff:fea5:b9e7 prefixlen 64 scopeid 0x20<link>
    ether 00:15:5d:a5:b9:e7 txqueuelen 1000 (Ethernet)
    RX packets 22 bytes 4550 (4.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 22 bytes 1548 (1.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

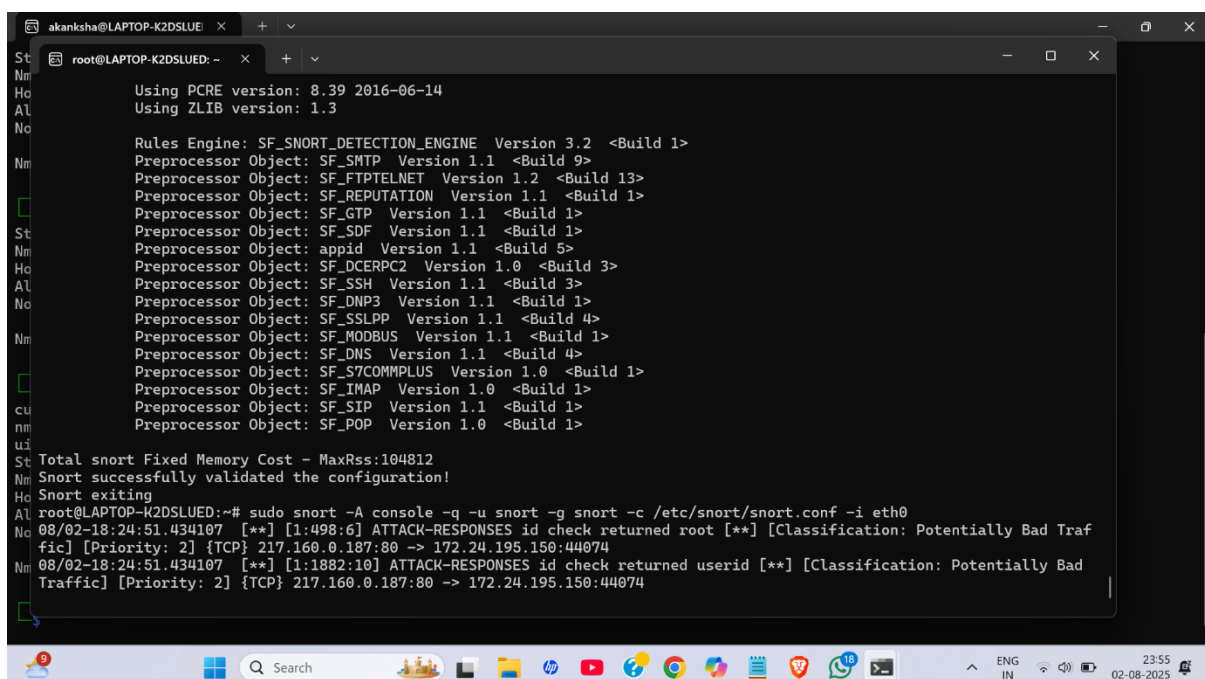
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 8 bytes 1631 (1.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 1631 (1.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(akanksha@LAPTOP-K2DSLUE)-[~]
$ nmap 172.24.195.150
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-02 21:32 IST
Nmap scan report for 172.24.195.150 (172.24.195.150)
Host is up (0.0000010s latency).
All 1000 scanned ports on 172.24.195.150 (172.24.195.150) are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 1 IP address (1 host up) scanned in 0.10 seconds

(akanksha@LAPTOP-K2DSLUE)-[~]
$
```

Figure 2.2: Nmap scan Snort successfully detected the scan and generated alerts.



```
akanksha@LAPTOP-K2DSLUE: ~
root@LAPTOP-K2DSLUE: ~
Using PCRE version: 8.39 2016-06-14
Using ZLIB version: 1.3

Rules Engine: SF_SNORT_DETECTION_ENGINE Version 3.2 <Build 1>
Preprocessor Object: SF_SMTP Version 1.1 <Build 9>
Preprocessor Object: SF_FTPTELNET Version 1.2 <Build 13>
Preprocessor Object: SF_REPUTATION Version 1.1 <Build 1>
Preprocessor Object: SF_GTP Version 1.1 <Build 1>
Preprocessor Object: SF_SDF Version 1.1 <Build 1>
Preprocessor Object: appid Version 1.1 <Build 5>
Preprocessor Object: SF_DCERPC2 Version 1.0 <Build 3>
Preprocessor Object: SF_SSH Version 1.1 <Build 3>
Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>
Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>
Preprocessor Object: SF_MODBUS Version 1.1 <Build 1>
Preprocessor Object: SF_DNS Version 1.1 <Build 4>
Preprocessor Object: SF_S7COMMPPLUS Version 1.0 <Build 1>
Preprocessor Object: SF_IMAP Version 1.0 <Build 1>
Preprocessor Object: SF_SIP Version 1.1 <Build 1>
Preprocessor Object: SF_POP Version 1.0 <Build 1>

Total snort Fixed Memory Cost - MaxRss:104812
Snort successfully validated the configuration!
Snort exiting

root@LAPTOP-K2DSLUE:~# sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
08/02-18:24:51.434107 [**] [1:498:6] ATTACK-RESPONSES id check returned root [**] [Classification: Potentially Bad Traf
fic] [Priority: 2] {TCP} 217.160.0.187:80 -> 172.24.195.150:44074
08/02-18:24:51.434107 [**] [1:1882:10] ATTACK-RESPONSES id check returned userid [**] [Classification: Potentially Bad
Traffic] [Priority: 2] {TCP} 217.160.0.187:80 -> 172.24.195.150:44074
```

Figure 2.3: Snort detecting the Nmap scan Legion is another tool for information gathering. This tool was employed to simulate more advanced reconnaissance and network mapping attacks. Using Legion, the IP address of the Ubuntu machine was scoped, and detailed scans were performed.

The attack targeted different services running on the Ubuntu machine, attempting to map open ports and exploit known vulnerabilities.

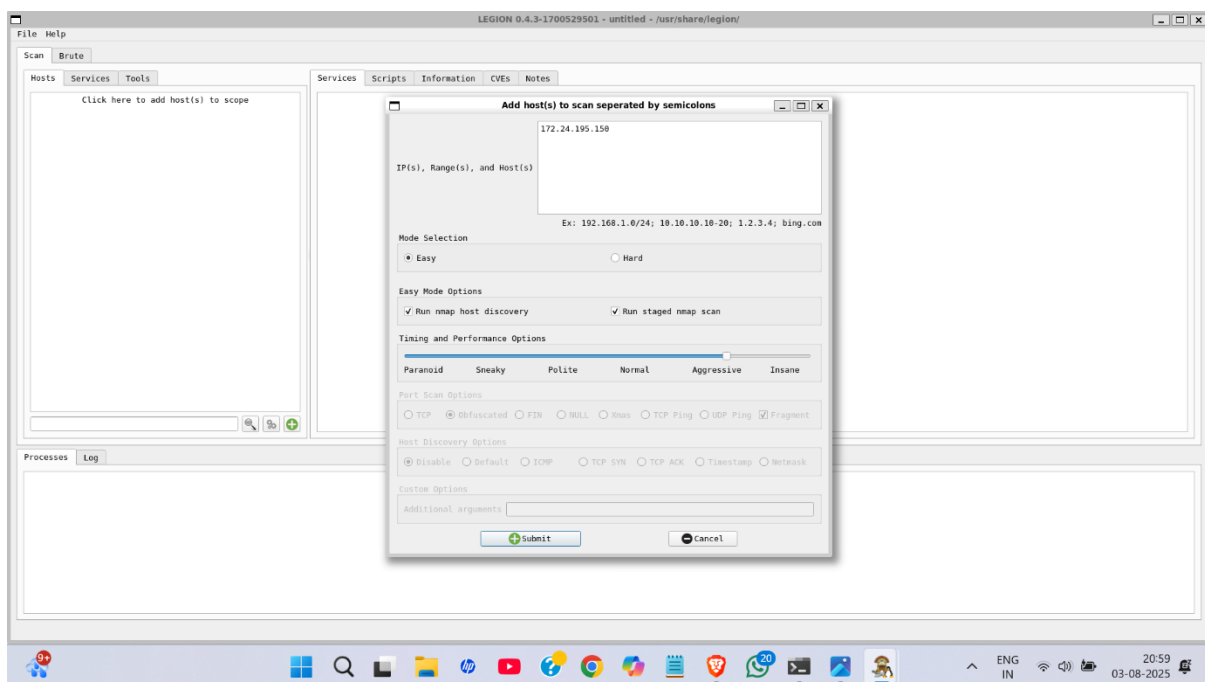
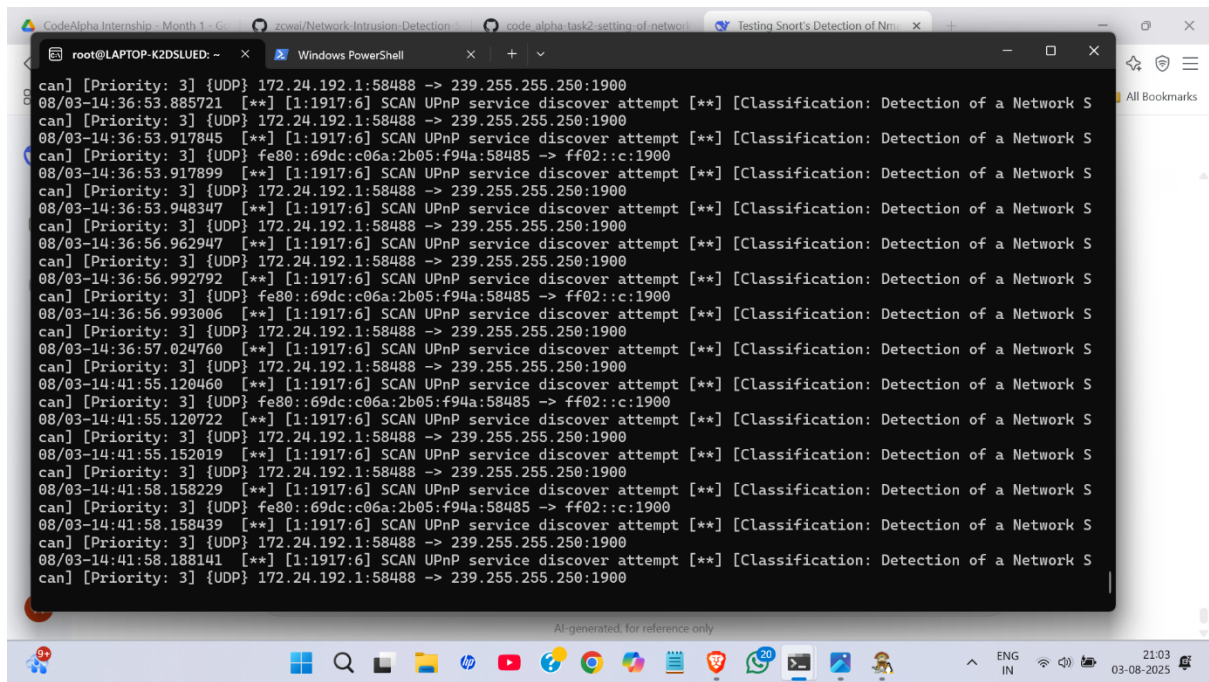


Figure 2.4: adding Ubuntu machine IP address to scope Snort was again successful in detecting these malicious activities and issued real-time alerts



```
can] [Priority: 3] {UDP} 172.24.192.1:58488 -> 239.255.255.250:1900
08/03-14:36:53.885721 [**] [1:1917:6] SCAN UPnP service discover attempt [**] [Classification: Detection of a Network S
can] [Priority: 3] {UDP} 172.24.192.1:58488 -> 239.255.255.250:1900
08/03-14:36:53.917845 [**] [1:1917:6] SCAN UPnP service discover attempt [**] [Classification: Detection of a Network S
can] [Priority: 3] {UDP} fe80::69dc:c06a:2b05:f94a:58485 -> ff02::c:1900
08/03-14:36:53.917899 [**] [1:1917:6] SCAN UPnP service discover attempt [**] [Classification: Detection of a Network S
can] [Priority: 3] {UDP} 172.24.192.1:58488 -> 239.255.255.250:1900
08/03-14:36:53.948347 [**] [1:1917:6] SCAN UPnP service discover attempt [**] [Classification: Detection of a Network S
can] [Priority: 3] {UDP} 172.24.192.1:58488 -> 239.255.255.250:1900
08/03-14:36:56.962947 [**] [1:1917:6] SCAN UPnP service discover attempt [**] [Classification: Detection of a Network S
can] [Priority: 3] {UDP} 172.24.192.1:58488 -> 239.255.255.250:1900
08/03-14:36:56.992792 [**] [1:1917:6] SCAN UPnP service discover attempt [**] [Classification: Detection of a Network S
can] [Priority: 3] {UDP} fe80::69dc:c06a:2b05:f94a:58485 -> ff02::c:1900
08/03-14:36:56.993006 [**] [1:1917:6] SCAN UPnP service discover attempt [**] [Classification: Detection of a Network S
can] [Priority: 3] {UDP} 172.24.192.1:58488 -> 239.255.255.250:1900
08/03-14:36:57.024760 [**] [1:1917:6] SCAN UPnP service discover attempt [**] [Classification: Detection of a Network S
can] [Priority: 3] {UDP} 172.24.192.1:58488 -> 239.255.255.250:1900
08/03-14:41:55.120460 [**] [1:1917:6] SCAN UPnP service discover attempt [**] [Classification: Detection of a Network S
can] [Priority: 3] {UDP} fe80::69dc:c06a:2b05:f94a:58485 -> ff02::c:1900
08/03-14:41:55.120722 [**] [1:1917:6] SCAN UPnP service discover attempt [**] [Classification: Detection of a Network S
can] [Priority: 3] {UDP} 172.24.192.1:58488 -> 239.255.255.250:1900
08/03-14:41:55.152019 [**] [1:1917:6] SCAN UPnP service discover attempt [**] [Classification: Detection of a Network S
can] [Priority: 3] {UDP} 172.24.192.1:58488 -> 239.255.255.250:1900
08/03-14:41:58.158229 [**] [1:1917:6] SCAN UPnP service discover attempt [**] [Classification: Detection of a Network S
can] [Priority: 3] {UDP} fe80::69dc:c06a:2b05:f94a:58485 -> ff02::c:1900
08/03-14:41:58.158439 [**] [1:1917:6] SCAN UPnP service discover attempt [**] [Classification: Detection of a Network S
can] [Priority: 3] {UDP} 172.24.192.1:58488 -> 239.255.255.250:1900
08/03-14:41:58.188141 [**] [1:1917:6] SCAN UPnP service discover attempt [**] [Classification: Detection of a Network S
can] [Priority: 3] {UDP} 172.24.192.1:58488 -> 239.255.255.250:1900
```

Figure 2.5: Snort detecting Legion scanning attack