Wazuh - Snort IDS

Wazuh-Snort IDS Integration Guide

Created By: Amir Raza

Follow Me: www.linkedin.com/in/amirsoc

Step-by-Step Guide:Integrating Wazuh and Snort IDS for Enhanced Security Monitoring

Introduction

Snort is a powerful and widely used open-source Intrusion Detection System (IDS). It monitors network traffic in real time and looks for suspicious patterns, policy violations, or known attacks using a set of rules. When Snort finds something suspicious, it generates alerts to help security teams take action.

Wazuh, on the other hand, is a modern open-source security monitoring platform. It combines several important features like intrusion detection, log analysis, vulnerability detection, and file integrity monitoring. Wazuh uses agents installed on different systems to collect and forward data to a central server, where this information is analyzed and visualized.

Why integrate Snort with Wazuh?

By integrating Snort with Wazuh, we bring together powerful network-based detection (from Snort) and advanced log analysis and visualization (from Wazuh). This offers:

Centralized Monitoring: All alerts generated by Snort are collected and viewed in a single dashboard within Wazuh, simplifying security management.
Better Visibility: Security teams gain a unified view of both network-based and host-based security alerts, improving overall situational awareness.Scalability: Wazuh can process alerts from multiple Snort sensors deployed across the network, making it easier to monitor large or distributed environments.
Improved Incident Response: Real-time alerts enable faster detection and quicker response to potential threats.
Customizable Rules and Alerts: Both Snort and Wazuh support customization, allowing fine-tuning of detection rules and alert priorities to suit organizational needs.

Use Cases

Incident Response: Rapidly detect and investigate suspicious network activities to contain threats before they cause damage.
Compliance Monitoring: Collect and organize logs and alerts to meet regulatory and industry compliance requirements such as PCI-DSS, HIPAA, or GDPR.
Threat Hunting: Proactively search through historical and real-time alerts to identify hidden, stealthy, or advanced persistent threats (APTs).

Network Security Monitoring: Continuously monitor network traffic to detect intrusions, policy violations, malware infections, and suspicious behaviors.
Security Operations Center (SOC) Support: Enhance SOC capabilities by providing comprehensive network and endpoint security data in one place.

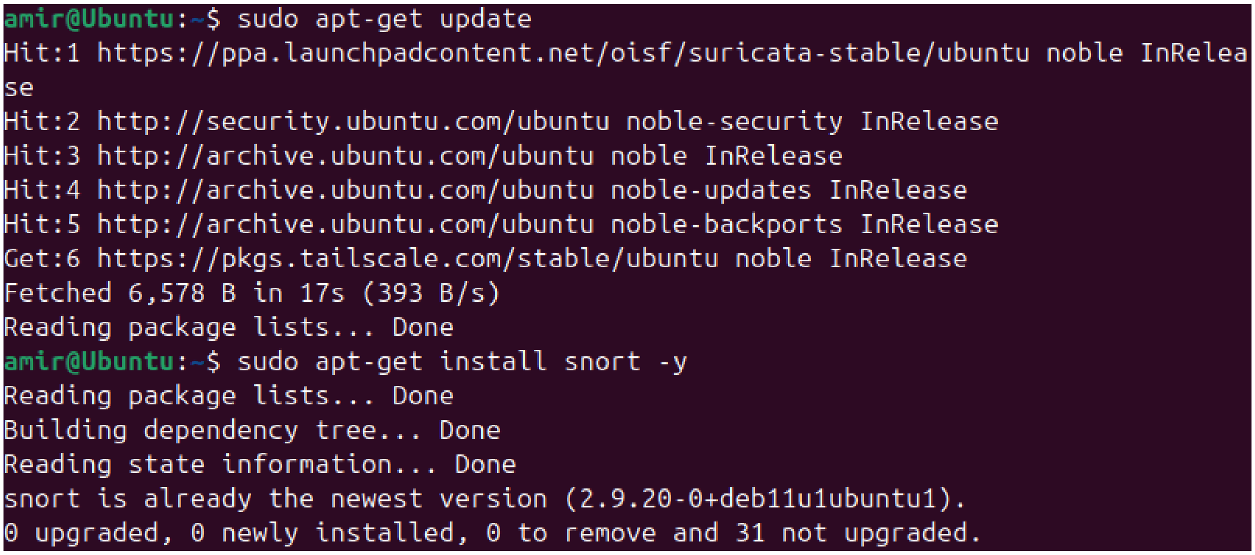
How the Integration Works

Snort monitors network traffic using its rules and generates alerts for suspicious events.
Wazuh agent on the Snort host reads Snort's alert logs and forwards them to the Wazuh manager.
Wazuh manager processes and correlates these alerts with data from other sources like endpoint agents, firewall logs, and system events.
Security analysts use the Wazuh dashboard to investigate alerts, prioritize incidents, and take action.

Install Snort on the Ubuntu Machine

To begin, update your package lists and then install Snort using the following commands:

```
sudo apt-get update                              sudo apt-get install snort -y
```

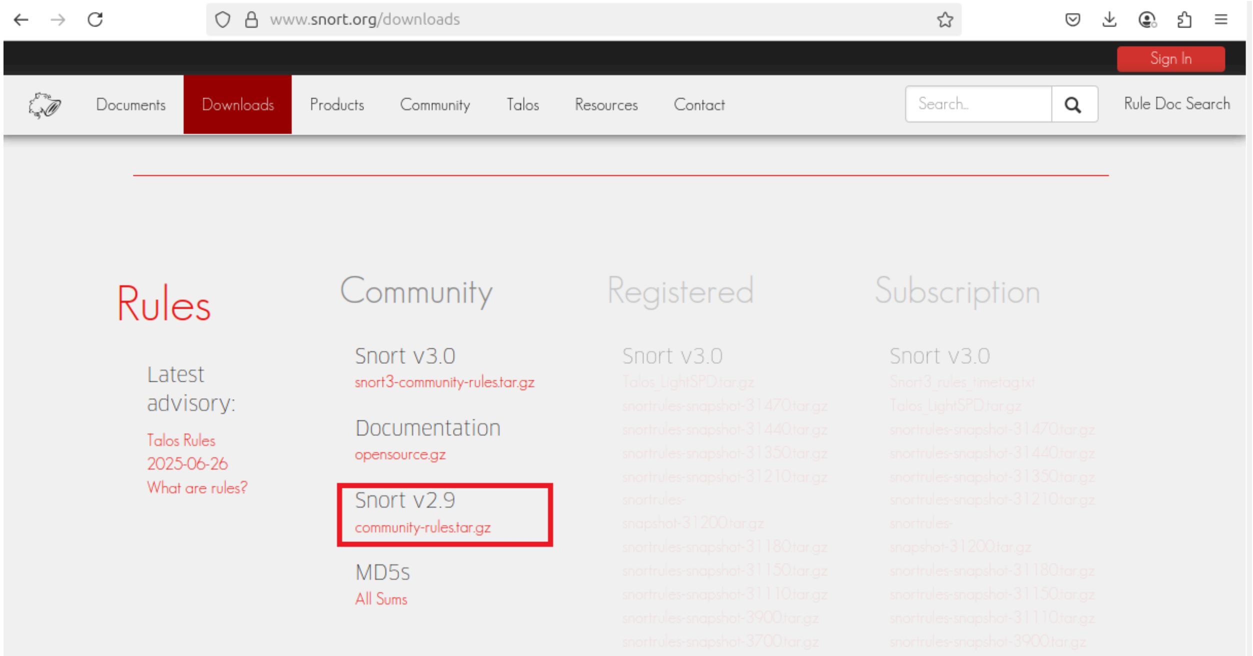


```
amir@Ubuntu:~$ sudo apt-get update
Hit:1 https://ppa.launchpadcontent.net/oisf/suricata-stable/ubuntu noble InRelea
se
Hit:2 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:3 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:4 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Get:6 https://pkgs.tailscale.com/stable/ubuntu noble InRelease
Fetched 6,578 B in 17s (393 B/s)
Reading package lists... Done
amir@Ubuntu:~$ sudo apt-get install snort -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
snort is already the newest version (2.9.20-0+deb11u1ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 31 not upgraded.
```

This ensures you have the latest package information and installs Snort IDS on your system so it's ready for configuration.

After installing Snort, download the free *Community Ruleset* from Snort's official website.

https://www.snort.org/downloads



Rules

Latest advisory:
Talos Rules 2025-06-26
What are rules?

Community

Snort v3.0
snort3-community-rules.tar.gz

Documentation
opensource.gz

Snort v2.9
community-rules.tar.gz

MD5s
All Sums

Registered

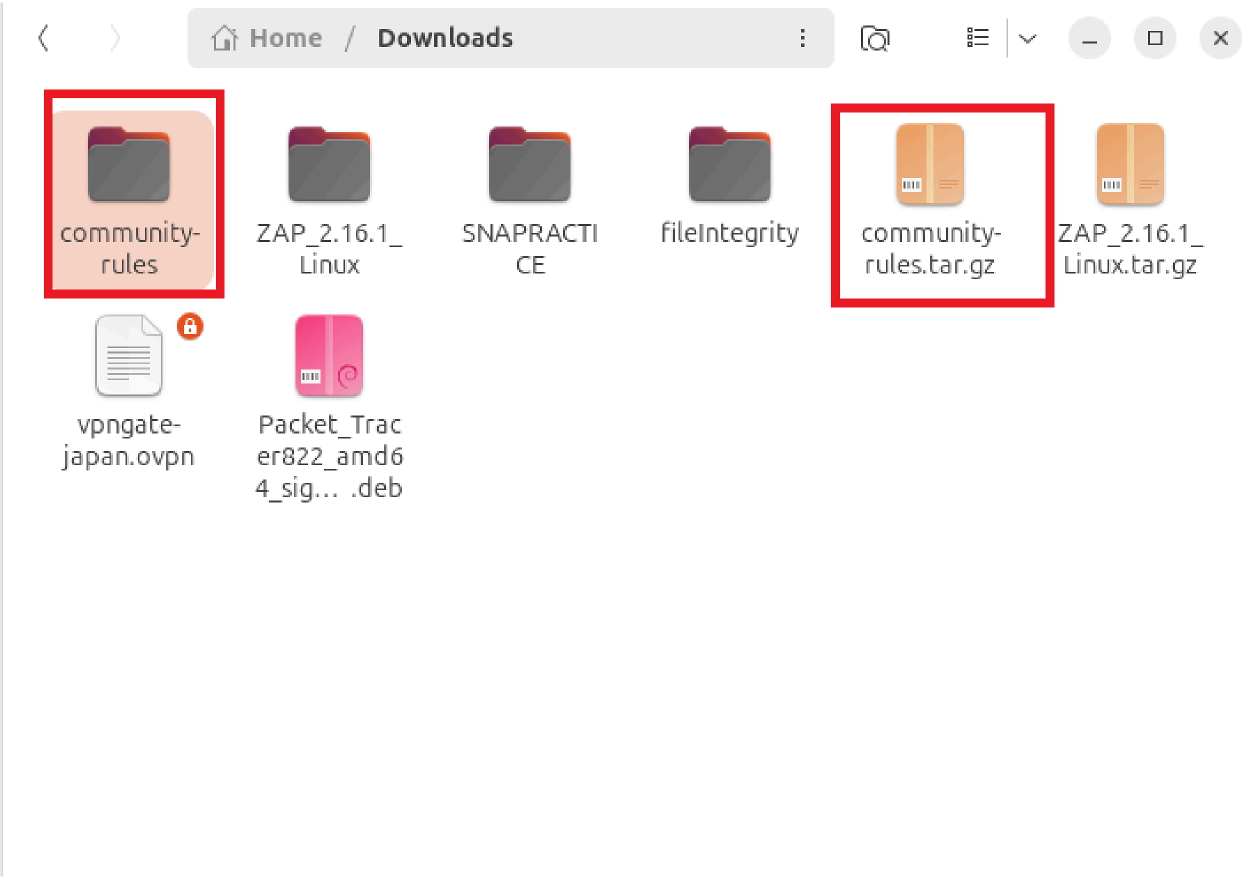
Snort v3.0
Talos_LightSPD.tar.gz
snortrules-snapshot-31470.tar.gz
snortrules-snapshot-31440.tar.gz
snortrules-snapshot-31350.tar.gz
snortrules-snapshot-31210.tar.gz
snortrules-snapshot-31200.tar.gz
snortrules-snapshot-31180.tar.gz
snortrules-snapshot-31150.tar.gz
snortrules-snapshot-31110.tar.gz
snortrules-snapshot-3900.tar.gz
snortrules-snapshot-3700.tar.gz

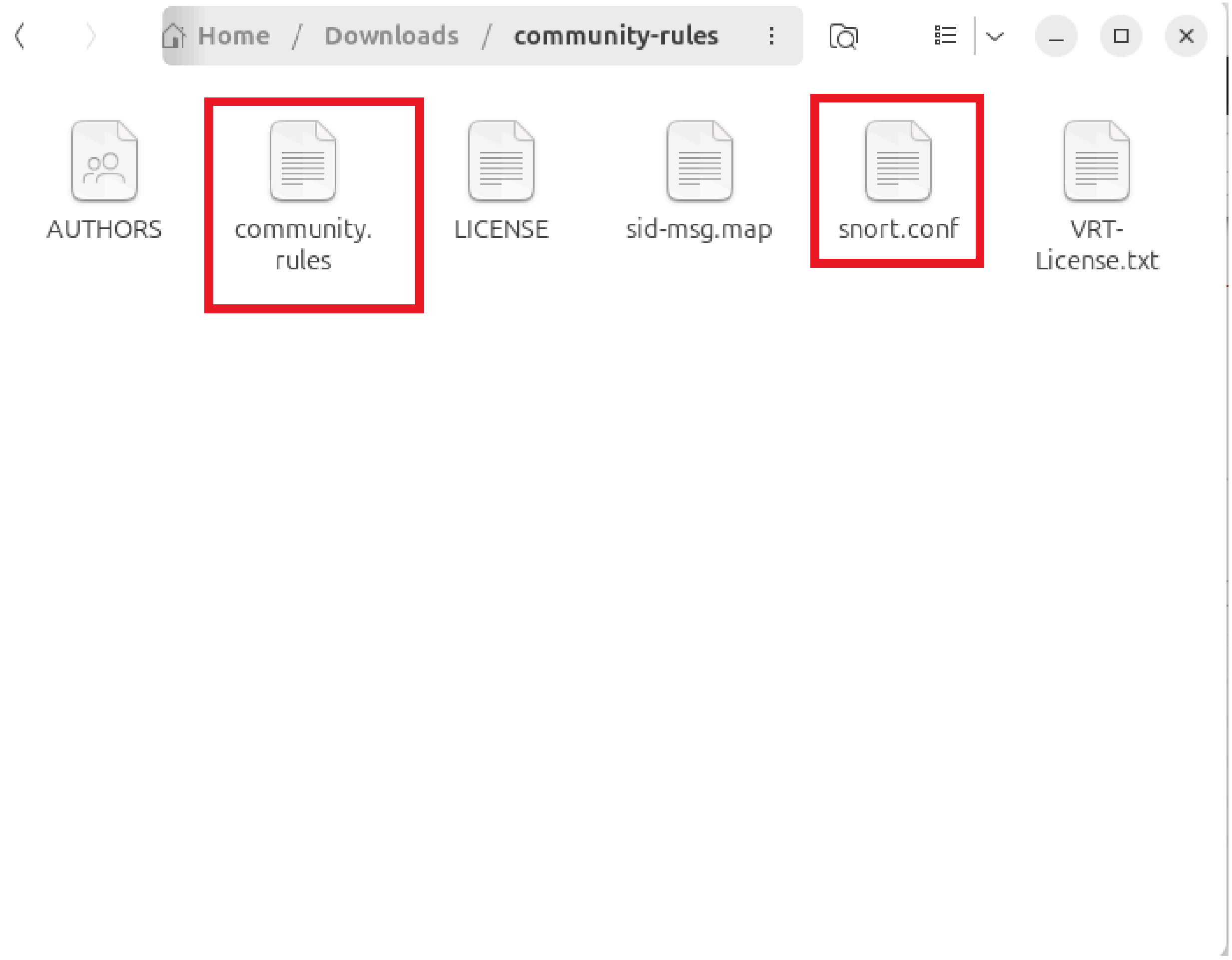
Subscription

Snort v3.0
Snort3_rules_timetag.txt
Talos_LightSPD.tar.gz
snortrules-snapshot-31470.tar.gz
snortrules-snapshot-31440.tar.gz
snortrules-snapshot-31350.tar.gz
snortrules-snapshot-31210.tar.gz
snortrules-snapshot-31200.tar.gz
snortrules-snapshot-31180.tar.gz
snortrules-snapshot-31150.tar.gz
snortrules-snapshot-31110.tar.gz
snortrules-snapshot-3900.tar.gz

After downloading the community-rules.tar.gz file, you need to extract its contents to the Snort rules directory.

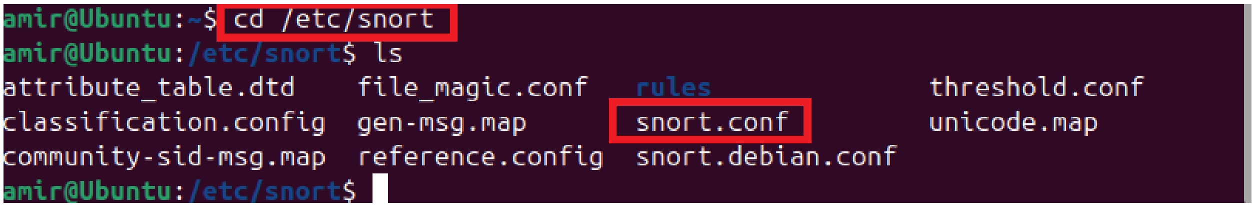
The extracted community.rules file contains the set of Snort detection rules. We will configure Snort to use this file in the upcoming steps.


Home / Downloads
community-rules
ZAP_2.16.1_Linux
SNAPRACTICE
fileIntegrity
community-rules.tar.gz
ZAP_2.16.1_Linux.tar.gz
vpngate-japan.ovpn
Packet_Tracer822_amd64_sig… .deb

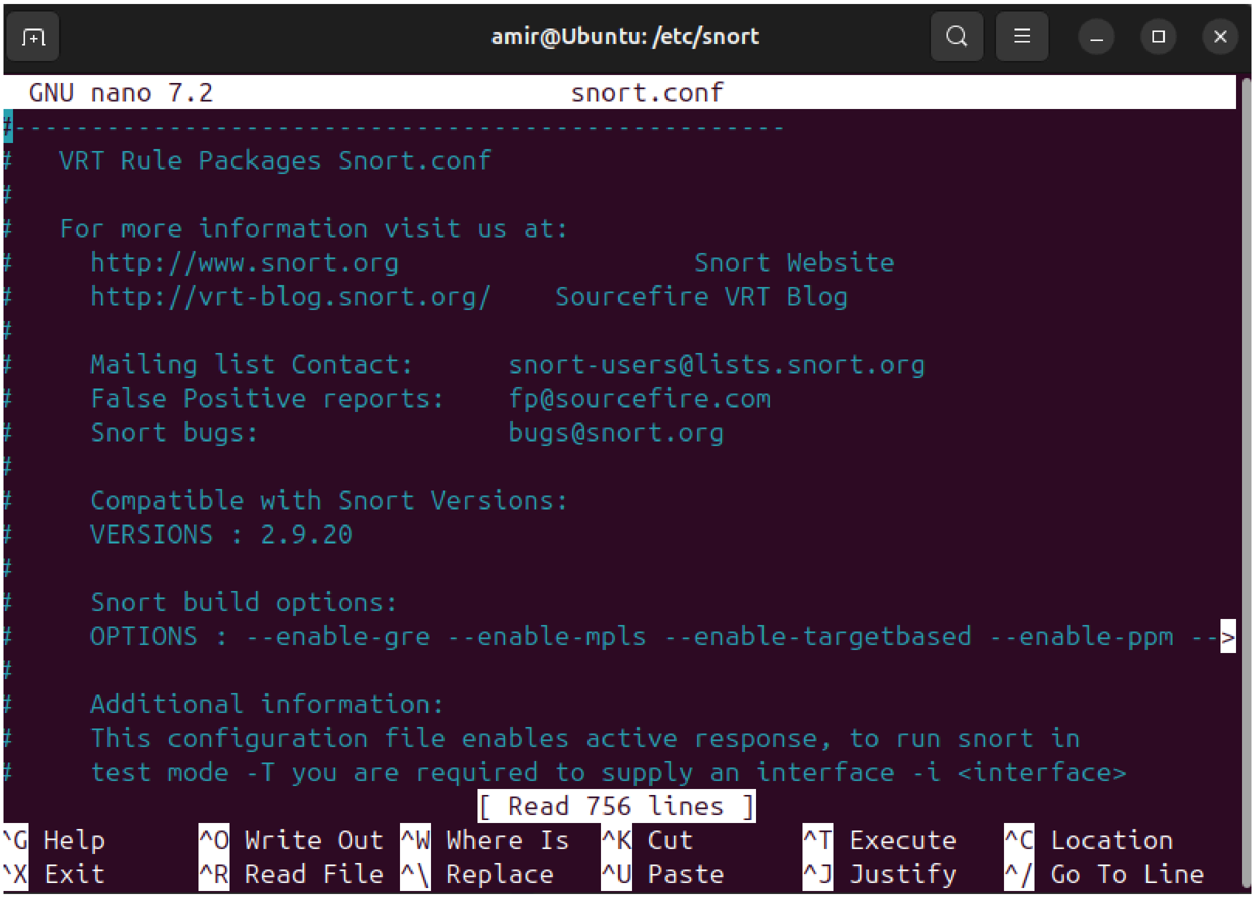


Next, navigate to the Snort configuration directory and edit the `snort.conf` file to customize Snort settings for your network environment.

cd /etc/snort   sudo nano snort.conf



```
amir@Ubuntu:~$ cd /etc/snort
amir@Ubuntu:/etc/snort$ ls
attribute_table.dtd    file_magic.conf   rules              threshold.conf
classification.config  gen-msg.map       snort.conf         unicode.map
community-sid-msg.map  reference.config  snort.debian.conf
amir@Ubuntu:/etc/snort$
```



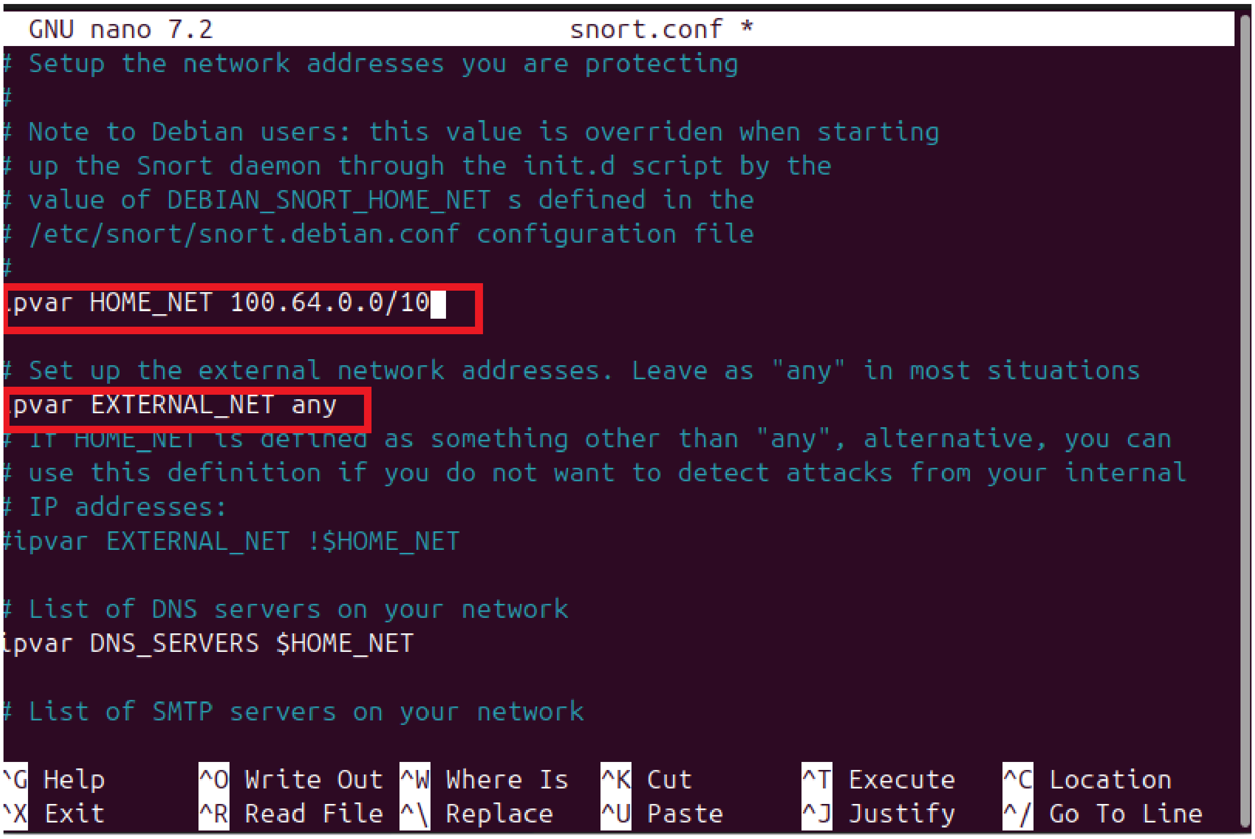
```
  GNU nano 7.2                          snort.conf
#--------------------------------------------------
#   VRT Rule Packages Snort.conf
#
#   For more information visit us at:
#     http://www.snort.org                   Snort Website
#     http://vrt-blog.snort.org/    Sourcefire VRT Blog
#
#     Mailing list Contact:      snort-users@lists.snort.org
#     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-targetbased --enable-ppm -->
#
#     Additional information:
#     This configuration file enables active response, to run snort in
#     test mode -T you are required to supply an interface -i <interface>
                               [ Read 756 lines ]
^G Help        ^O Write Out ^W Where Is  ^K Cut        ^T Execute   ^C Location
^X Exit        ^R Read File ^\ Replace   ^U Paste      ^J Justify   ^/ Go To Line
```

Network Configuration

At this stage, you need to update the network settings in the Snort configuration file according to your specific environment. Carefully follow the provided example (or figure) but replace the details with those relevant to your network setup.

Important:

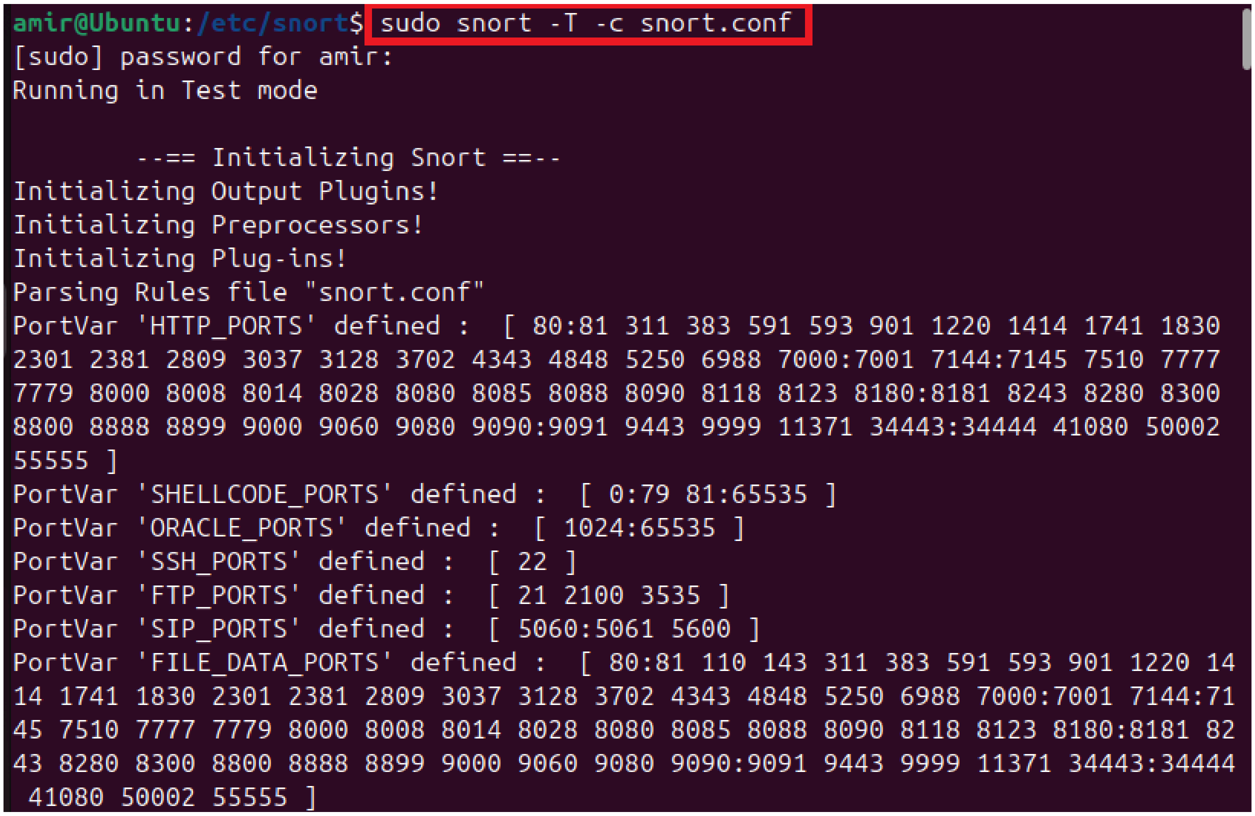
Make sure to specify the correct network interface that Snort should monitor. Scroll down in the configuration file and update the interface setting accordingly. This step is crucial for Snort to capture network traffic properly.



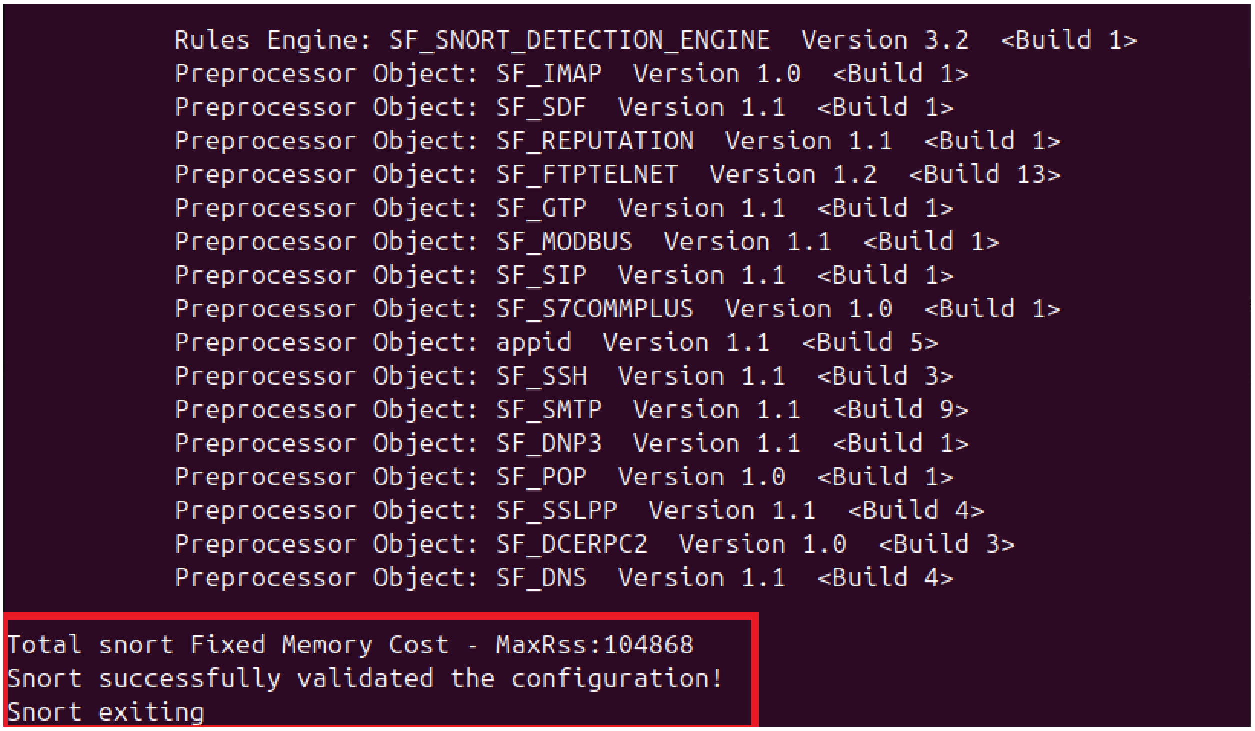
```
  GNU nano 7.2                          snort.conf *
# Setup the network addresses you are protecting
#
# Note to Debian users: this value is overriden when starting
# up the Snort daemon through the init.d script by the
# value of DEBIAN_SNORT_HOME_NET s defined in the
# /etc/snort/snort.debian.conf configuration file
#
ipvar HOME_NET 100.64.0.0/10

# Set up the external network addresses. Leave as "any" in most situations
ipvar EXTERNAL_NET any
# If HOME_NET is defined as something other than "any", alternative, you can
# use this definition if you do not want to detect attacks from your internal
# IP addresses:
#ipvar EXTERNAL_NET !$HOME_NET

# List of DNS servers on your network
ipvar DNS_SERVERS $HOME_NET

# List of SMTP servers on your network

^G Help        ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit        ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line
```

After editing the `snort.conf` file, it's important to check that your configuration is valid before running Snort.

```
sudo snort -T -c snort.conf
```



```
amir@Ubuntu:/etc/snort$ sudo snort -T -c snort.conf
[sudo] password for amir:
Running in Test mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "snort.conf"
PortVar 'HTTP_PORTS' defined :  [ 80:81 311 383 591 593 901 1220 1414 1741 1830
2301 2381 2809 3037 3128 3702 4343 4848 5250 6988 7000:7001 7144:7145 7510 7777
7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:8181 8243 8280 8300
8800 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 34443:34444 41080 50002
55555 ]
PortVar 'SHELLCODE_PORTS' defined :  [ 0:79 81:65535 ]
PortVar 'ORACLE_PORTS' defined :  [ 1024:65535 ]
PortVar 'SSH_PORTS' defined :  [ 22 ]
PortVar 'FTP_PORTS' defined :  [ 21 2100 3535 ]
PortVar 'SIP_PORTS' defined :  [ 5060:5061 5600 ]
PortVar 'FILE_DATA_PORTS' defined :  [ 80:81 110 143 311 383 591 593 901 1220 14
14 1741 1830 2301 2381 2809 3037 3128 3702 4343 4848 5250 6988 7000:7001 7144:71
45 7510 7777 7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:8181 82
43 8280 8300 8800 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 34443:34444
 41080 50002 55555 ]
```



```
           Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 3.2  <Build 1>
           Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
           Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
           Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
           Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
           Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
           Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
           Preprocessor Object: SF_S7COMMPLUS  Version 1.0  <Build 1>
           Preprocessor Object: appid  Version 1.1  <Build 5>
           Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
           Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
           Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
           Preprocessor Object: SF_POP  Version 1.0  <Build 1>
           Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
           Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>
           Preprocessor Object: SF_DNS  Version 1.1  <Build 4>

Total snort Fixed Memory Cost - MaxRss:104868
Snort successfully validated the configuration!
Snort exiting
```

Now snort configure successfully.

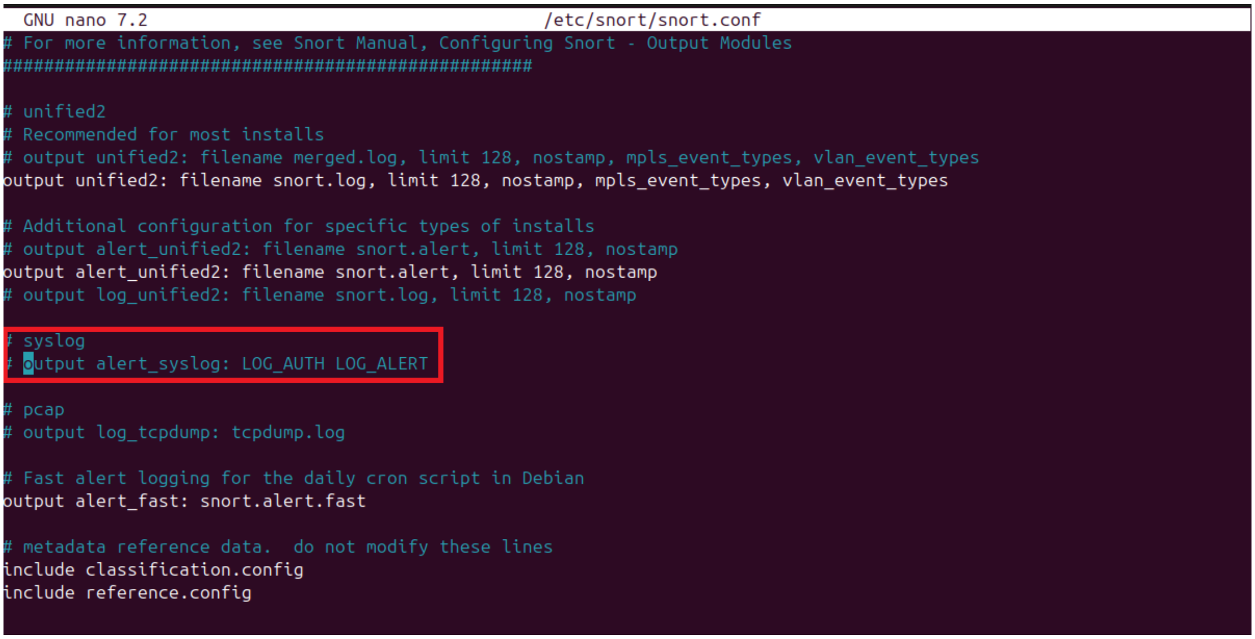
Enable Syslog Logging for Snort and Integrate with Wazuh

Edit `snort.conf` file

Open `/etc/snort/snort.conf.`

Uncomment syslog output line

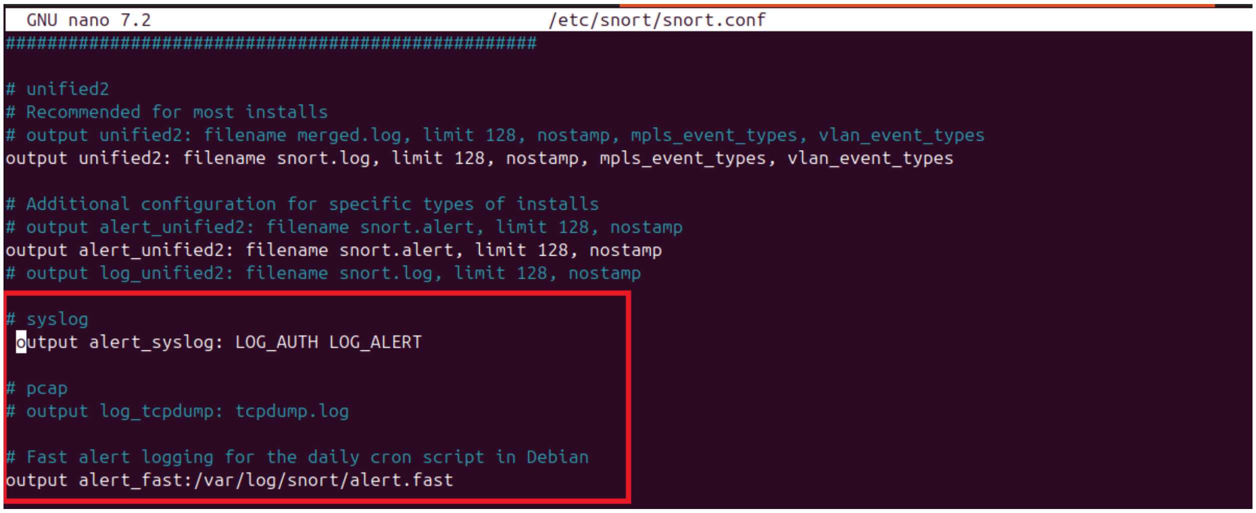
Find the line: `# output alert_syslog: LOG_AUTH LOG_ALERT and remove #.`



```
  GNU nano 7.2                                          /etc/snort/snort.conf
# For more information, see Snort Manual, Configuring Snort - Output Modules
###################################################

# unified2
# Recommended for most installs
# output unified2: filename merged.log, limit 128, nostamp, mpls_event_types, vlan_event_types
output unified2: filename snort.log, limit 128, nostamp, mpls_event_types, vlan_event_types

# Additional configuration for specific types of installs
# output alert_unified2: filename snort.alert, limit 128, nostamp
output alert_unified2: filename snort.alert, limit 128, nostamp
# output log_unified2: filename snort.log, limit 128, nostamp

# syslog
# output alert_syslog: LOG_AUTH LOG_ALERT

# pcap
# output log_tcpdump: tcpdump.log

# Fast alert logging for the daily cron script in Debian
output alert_fast: snort.alert.fast

# metadata reference data.  do not modify these lines
include classification.config
include reference.config
```

And also change and add this.

Output alert_fast:/var/log/snort/alert.fast



```
  GNU nano 7.2                                          /etc/snort/snort.conf
###################################################

# unified2
# Recommended for most installs
# output unified2: filename merged.log, limit 128, nostamp, mpls_event_types, vlan_event_types
output unified2: filename snort.log, limit 128, nostamp, mpls_event_types, vlan_event_types

# Additional configuration for specific types of installs
# output alert_unified2: filename snort.alert, limit 128, nostamp
output alert_unified2: filename snort.alert, limit 128, nostamp
# output log_unified2: filename snort.log, limit 128, nostamp

# syslog
 output alert_syslog: LOG_AUTH LOG_ALERT

# pcap
# output log_tcpdump: tcpdump.log

# Fast alert logging for the daily cron script in Debian
output alert_fast:/var/log/snort/alert.fast
```

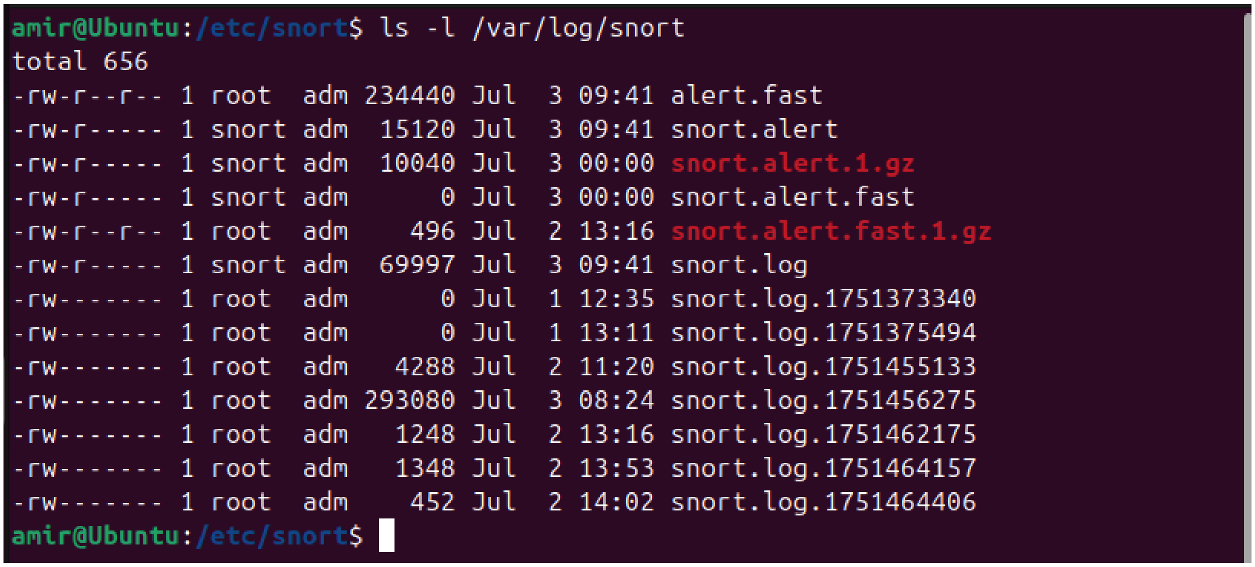
Save and exit.

Configure Wazuh Agent to Monitor Snort Logs

Check Current Permissions

You can check the current permissions and ownership of the Snort log files using the ls -l command:

```
ls -l /var/log/snort
```



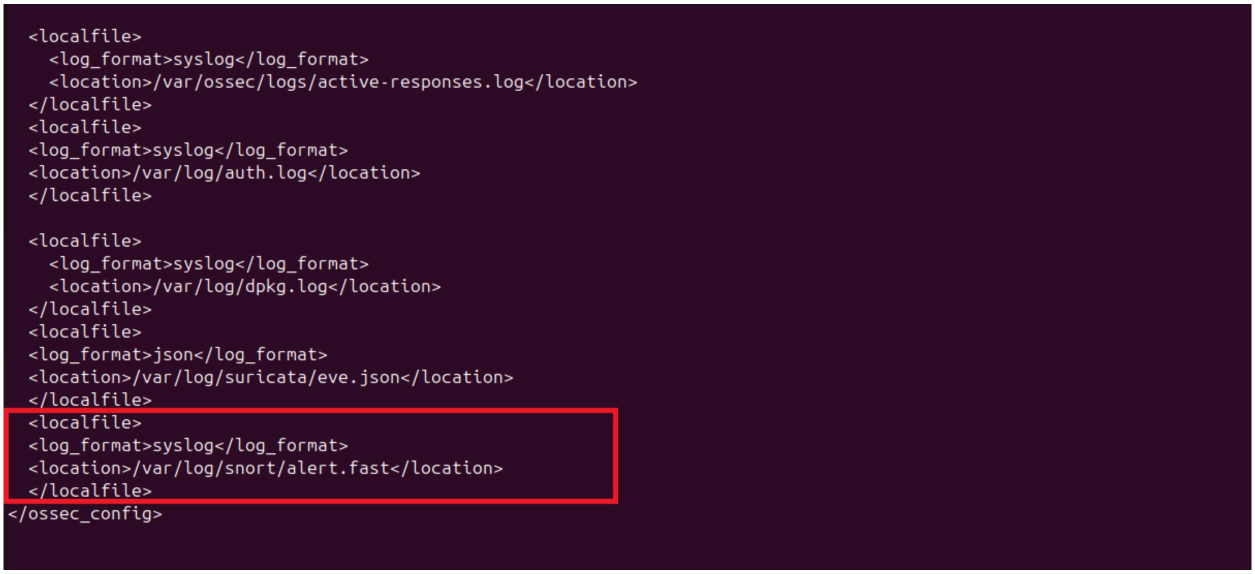
```
amir@Ubuntu:/etc/snort$ ls -l /var/log/snort
total 656
-rw-r--r-- 1 root  adm 234440 Jul  3 09:41 alert.fast
-rw-r----- 1 snort adm  15120 Jul  3 09:41 snort.alert
-rw-r----- 1 snort adm  10040 Jul  3 00:00 snort.alert.1.gz
-rw-r----- 1 snort adm      0 Jul  3 00:00 snort.alert.fast
-rw-r--r-- 1 root  adm    496 Jul  2 13:16 snort.alert.fast.1.gz
-rw-r----- 1 snort adm  69997 Jul  3 09:41 snort.log
-rw------- 1 root  adm      0 Jul  1 12:35 snort.log.1751373340
-rw------- 1 root  adm      0 Jul  1 13:11 snort.log.1751375494
-rw------- 1 root  adm   4288 Jul  2 11:20 snort.log.1751455133
-rw------- 1 root  adm 293080 Jul  3 08:24 snort.log.1751456275
-rw------- 1 root  adm   1248 Jul  2 13:16 snort.log.1751462175
-rw------- 1 root  adm   1348 Jul  2 13:53 snort.log.1751464157
-rw------- 1 root  adm    452 Jul  2 14:02 snort.log.1751464406
amir@Ubuntu:/etc/snort$ 
```

Open the Wazuh agent configuration file:

```
sudo nano /var/ossec/etc/ossec.conf
```

Inside the `<ossec_config>` block (but **outside** other tags like `<rules>` or `<syscheck>`), add the following:

```
<localfile>

  <log_format>syslog</log_format>

  <location>/var/log/snort/alert.fast</location>

</localfile>
```

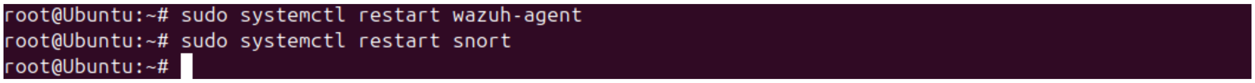


```
  <localfile>
    <log_format>syslog</log_format>
    <location>/var/ossec/logs/active-responses.log</location>
  </localfile>
  <localfile>
  <log_format>syslog</log_format>
  <location>/var/log/auth.log</location>
  </localfile>

  <localfile>
    <log_format>syslog</log_format>
    <location>/var/log/dpkg.log</location>
  </localfile>
  <localfile>
  <log_format>json</log_format>
  <location>/var/log/suricata/eve.json</location>
  </localfile>
  <localfile>
  <log_format>syslog</log_format>
  <location>/var/log/snort/alert.fast</location>
  </localfile>
</ossec_config>
```

Restart Both Services

sudo systemctl restart wazuh-agent

Sudo systemctl restart snort



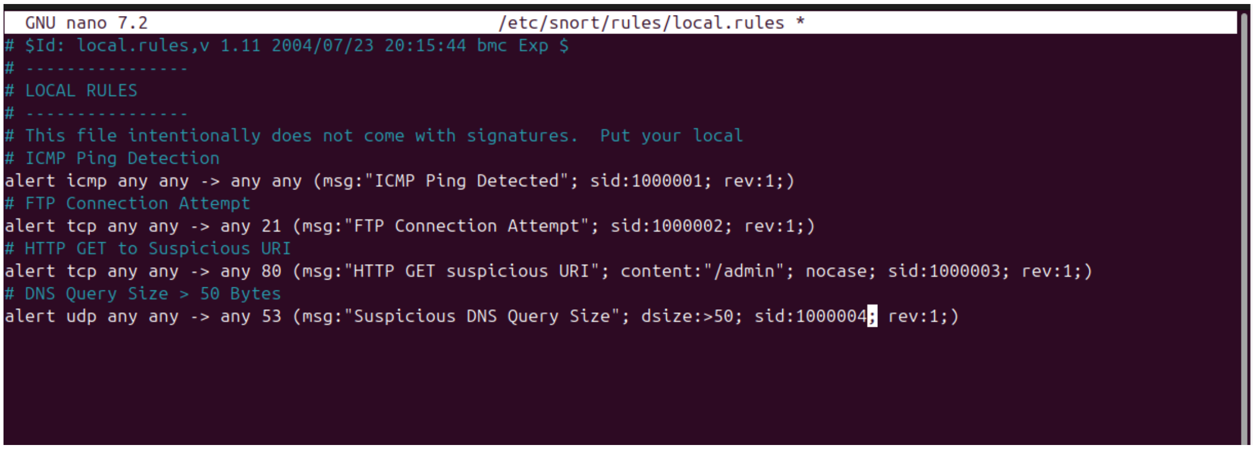
```
root@Ubuntu:~# sudo systemctl restart wazuh-agent
root@Ubuntu:~# sudo systemctl restart snort
root@Ubuntu:~#
```

Create & Add Snort Rules

Open the local Snort rules file:

sudo nano /etc/snort/rules/local.rules

And add following rules:

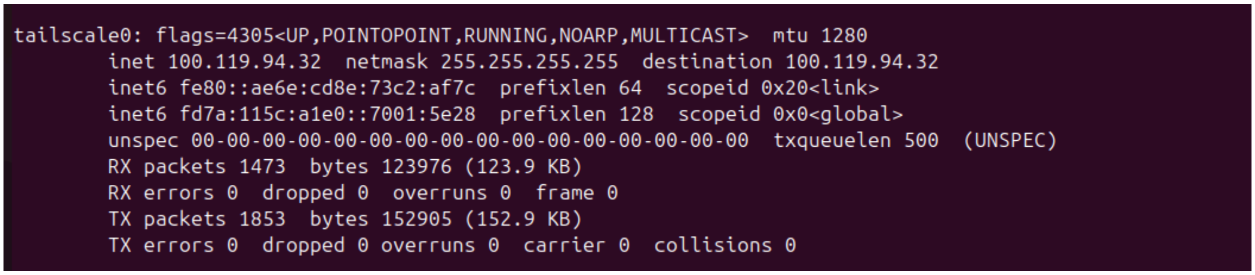


```
  GNU nano 7.2                              /etc/snort/rules/local.rules *
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# ----------------
# LOCAL RULES
# ----------------
# This file intentionally does not come with signatures.  Put your local
# ICMP Ping Detection
alert icmp any any -> any any (msg:"ICMP Ping Detected"; sid:1000001; rev:1;)
# FTP Connection Attempt
alert tcp any any -> any 21 (msg:"FTP Connection Attempt"; sid:1000002; rev:1;)
# HTTP GET to Suspicious URI
alert tcp any any -> any 80 (msg:"HTTP GET suspicious URI"; content:"/admin"; nocase; sid:1000003; rev:1;)
# DNS Query Size > 50 Bytes
alert udp any any -> any 53 (msg:"Suspicious DNS Query Size"; dsize:>50; sid:1000004; rev:1;)
```

Run Snort in IDS Mode

First, find your active network interface using:

ifconfig



```
tailscale0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1280
        inet 100.119.94.32  netmask 255.255.255.255  destination 100.119.94.32
        inet6 fe80::ae6e:cd8e:73c2:af7c  prefixlen 64  scopeid 0x20<link>
        inet6 fd7a:115c:a1e0::7001:5e28  prefixlen 128  scopeid 0x0<global>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
        RX packets 1473  bytes 123976 (123.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1853  bytes 152905 (152.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Then run Snort on that interface

sudo snort -A console -q -c /etc/snort/snort.conf -i tailscale0

Simulate Attacks (From Attacker Machine)

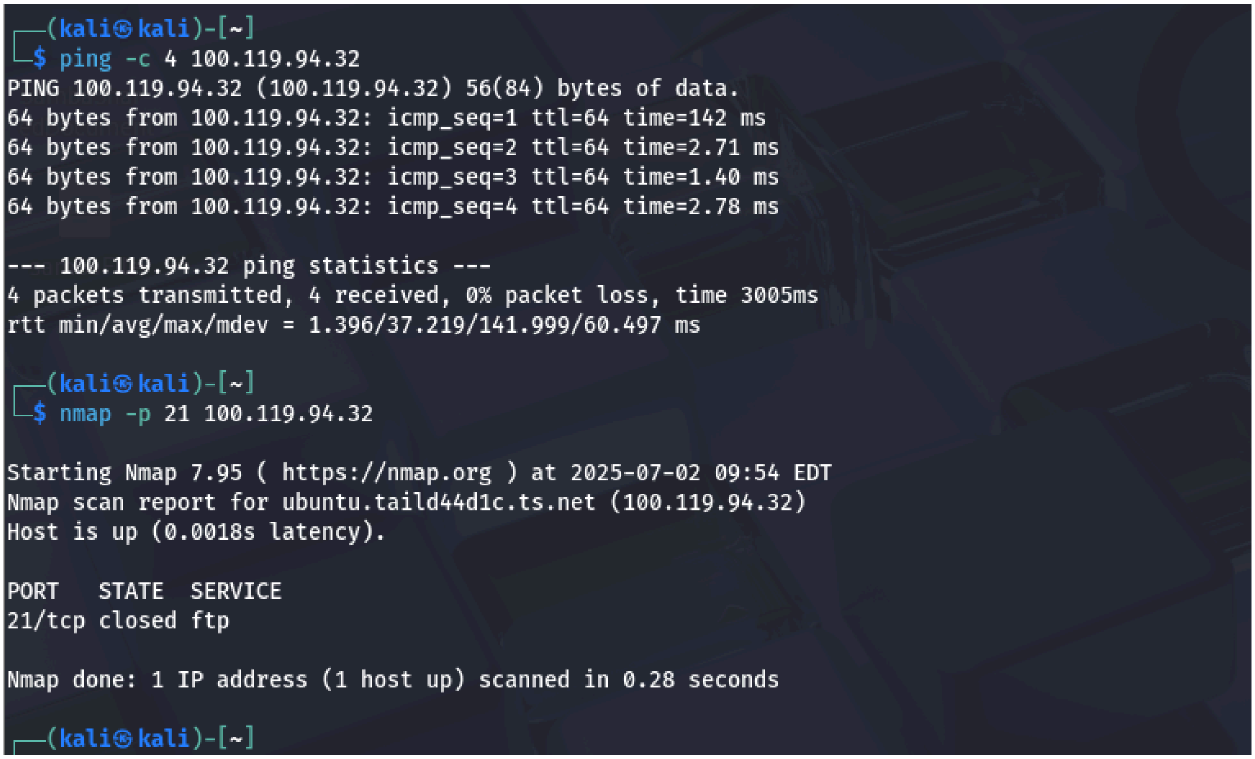
Use these commands from the attacker machine to generate traffic and trigger Snort alerts:

ICMP Ping:

ping -c 4 <Snort_IP>

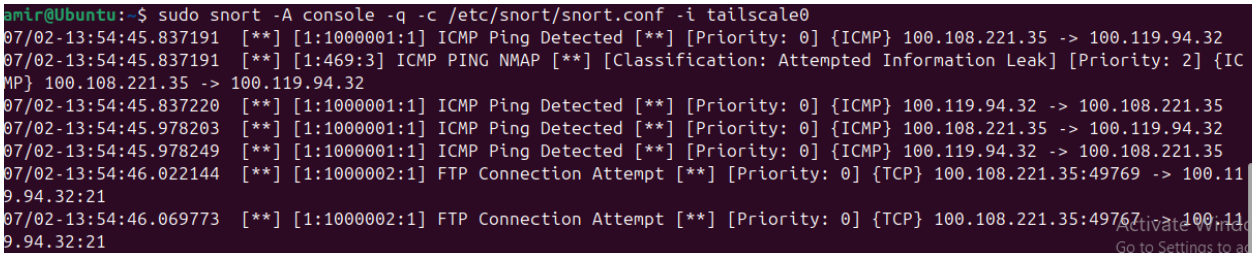
FTP Scan:

nmap -p 21 <Snort_IP>



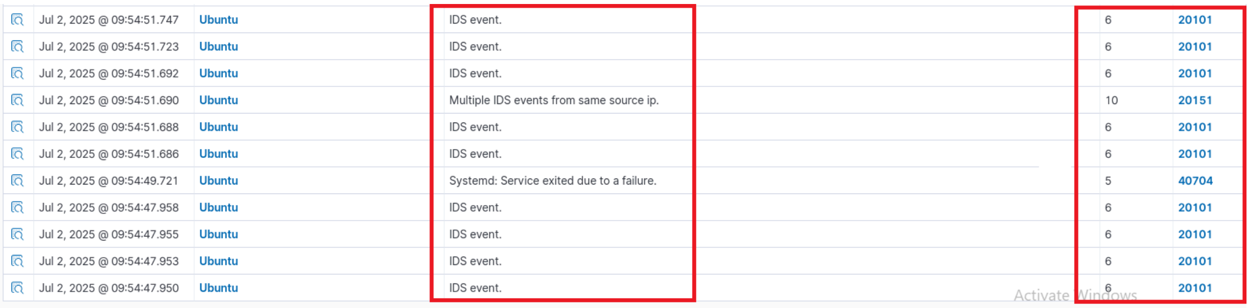
```
┌──(kali㉿kali)-[~]
└─$ ping -c 4 100.119.94.32
PING 100.119.94.32 (100.119.94.32) 56(84) bytes of data.
64 bytes from 100.119.94.32: icmp_seq=1 ttl=64 time=142 ms
64 bytes from 100.119.94.32: icmp_seq=2 ttl=64 time=2.71 ms
64 bytes from 100.119.94.32: icmp_seq=3 ttl=64 time=1.40 ms
64 bytes from 100.119.94.32: icmp_seq=4 ttl=64 time=2.78 ms

--- 100.119.94.32 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 1.396/37.219/141.999/60.497 ms

┌──(kali㉿kali)-[~]
└─$ nmap -p 21 100.119.94.32

Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-02 09:54 EDT
Nmap scan report for ubuntu.taild44d1c.ts.net (100.119.94.32)
Host is up (0.0018s latency).

PORT   STATE  SERVICE
21/tcp closed ftp

Nmap done: 1 IP address (1 host up) scanned in 0.28 seconds

┌──(kali㉿kali)-[~]
```

You’ll see alerts in real-time on the terminal if Snort is running with -A console.

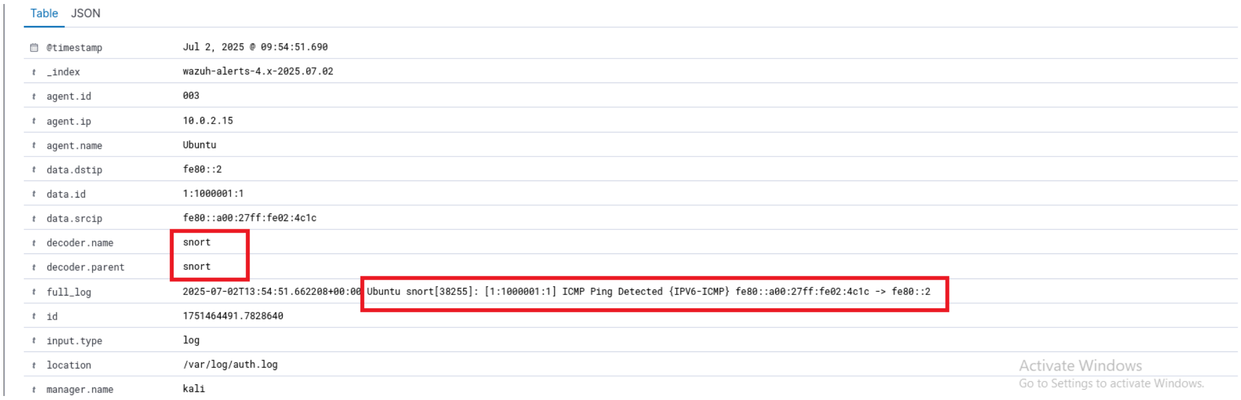


```
amir@Ubuntu:~$ sudo snort -A console -q -c /etc/snort/snort.conf -i tailscale0
07/02-13:54:45.837191  [**] [1:1000001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 100.108.221.35 -> 100.119.94.32
07/02-13:54:45.837191  [**] [1:469:3] ICMP PING NMAP [**] [Classification: Attempted Information Leak] [Priority: 2] {IC
MP} 100.108.221.35 -> 100.119.94.32
07/02-13:54:45.837220  [**] [1:1000001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 100.119.94.32 -> 100.108.221.35
07/02-13:54:45.978203  [**] [1:1000001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 100.108.221.35 -> 100.119.94.32
07/02-13:54:45.978249  [**] [1:1000001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 100.119.94.32 -> 100.108.221.35
07/02-13:54:46.022144  [**] [1:1000002:1] FTP Connection Attempt [**] [Priority: 0] {TCP} 100.108.221.35:49769 -> 100.11
9.94.32:21
07/02-13:54:46.069773  [**] [1:1000002:1] FTP Connection Attempt [**] [Priority: 0] {TCP} 100.108.221.35:49767 -> 100.11
9.94.32:21
```

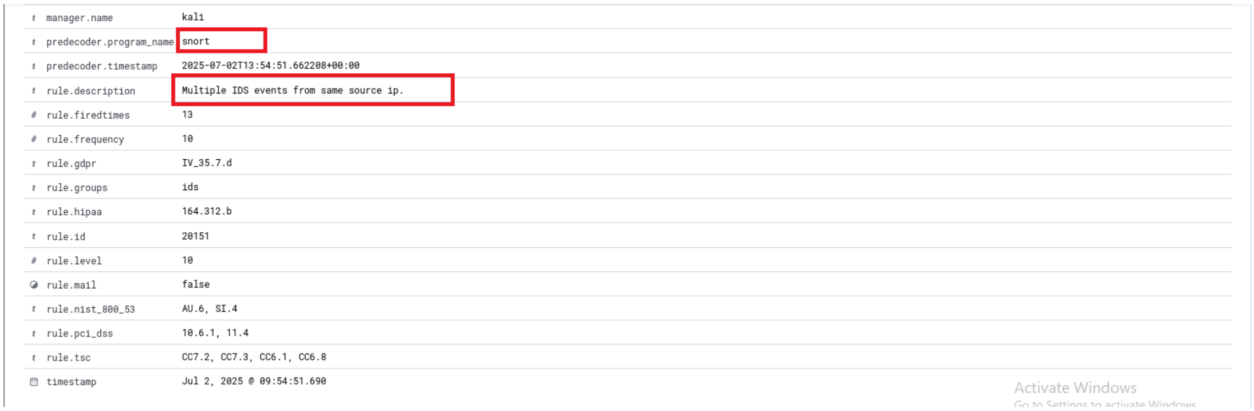
Access Wazuh Dashboard: Log in to the Wazuh web interface and navigate to the Security Events or Alerts section. Use filters to search for Snort-related alerts.



Jul 2, 2025 @ 09:54:51.747	Ubuntu	IDS event.	6	20101
Jul 2, 2025 @ 09:54:51.723	Ubuntu	IDS event.	6	20101
Jul 2, 2025 @ 09:54:51.692	Ubuntu	IDS event.	6	20101
Jul 2, 2025 @ 09:54:51.690	Ubuntu	Multiple IDS events from same source ip.	10	20151
Jul 2, 2025 @ 09:54:51.688	Ubuntu	IDS event.	6	20101
Jul 2, 2025 @ 09:54:51.686	Ubuntu	IDS event.	6	20101
Jul 2, 2025 @ 09:54:49.721	Ubuntu	Systemd: Service exited due to a failure.	5	40704
Jul 2, 2025 @ 09:54:47.958	Ubuntu	IDS event.	6	20101
Jul 2, 2025 @ 09:54:47.955	Ubuntu	IDS event.	6	20101
Jul 2, 2025 @ 09:54:47.953	Ubuntu	IDS event.	6	20101
Jul 2, 2025 @ 09:54:47.950	Ubuntu	IDS event.	6	20101



@timestamp	Jul 2, 2025 @ 09:54:51.690
_index	wazuh-alerts-4.x-2025.07.02
agent.id	003
agent.ip	10.0.2.15
agent.name	Ubuntu
data.dstip	fe80::2
data.id	1:1000001:1
data.srcip	fe80::a00:27ff:fe02:4c1c
decoder.name	snort
decoder.parent	snort
full_log	2025-07-02T13:54:51.662208+00:00 Ubuntu snort[38255]: [1:1000001:1] ICMP Ping Detected {IPV6-ICMP} fe80::a00:27ff:fe02:4c1c -> fe80::2
id	1751464491.7828640
input.type	log
location	/var/log/auth.log
manager.name	kali



manager.name	kali
predecoder.program_name	snort
predecoder.timestamp	2025-07-02T13:54:51.662208+00:00
rule.description	Multiple IDS events from same source ip.
rule.firedtimes	13
rule.frequency	10
rule.gdpr	IV_35.7.d
rule.groups	ids
rule.hipaa	164.312.b
rule.id	20151
rule.level	10
rule.mail	false
rule.nist_800_53	AU.6, SI.4
rule.pci_dss	10.6.1, 11.4
rule.tsc	CC7.2, CC7.3, CC6.1, CC6.8
timestamp	Jul 2, 2025 @ 09:54:51.690

Summary

In this task, I integrated Snort, an open-source intrusion detection system (IDS), with Wazuh SIEM to monitor and analyze network security events from a single dashboard.

I started by setting up Snort to detect different types of suspicious activities, like ICMP ping scans, FTP and SSH connection attempts, HTTP requests to sensitive URLs, and unusual DNS queries. I added custom rules to make sure Snort would generate alerts for these events.

After that, I configured the Wazuh Agent to read Snort's alert log files. These alerts were then sent to the Wazuh Manager, which collected and displayed them on the Wazuh dashboard. This way, I could easily see when Snort detected something suspicious on the network.

To test the setup, I simulated different attacks from an attacker machine, such as pinging, scanning FTP ports using Nmap, trying to access admin pages with curl, and making DNS queries. As expected, these activities triggered alerts in Snort, which were then forwarded to Wazuh and shown on the dashboard.

Conclusion

By following these steps, I successfully connected Snort with Wazuh SIEM. This integration makes it easier to keep an eye on network traffic and quickly spot any suspicious behavior from one central place. It helps improve the overall security by making it simpler to detect, investigate, and respond to potential threats.

Additionally, by checking that Snort's alert files exist, have the correct permissions, and making sure the Wazuh agent reads them correctly, I was able to make sure everything works smoothly without permission errors.

This setup is a practical way to enhance visibility and protect the network from possible intrusions.