Integration of Snort IDS with Wazuh!

SAMEER HASSAN

Wazuh lab

Github-link: GitHub - sameerhassancode/Wazuh-labs

Linkedin: https://www.linkedin.com/in/sameer-hassan-15a428255/

What is Wazuh?

Wazuh is a free and open-source security platform used for threat detection, compliance monitoring, and incident response. It helps organizations monitor their infrastructure in real-time by collecting and analyzing data from endpoints (like servers, desktops, or cloud instances). Wazuh works as a SIEM (Security Information and Event Management) and XDR (Extended Detection and Response) solution.

Snort

Snort is an open-source Intrusion Detection System (IDS) that monitors network traffic in real time to detect and prevent suspicious activities. Developed by Cisco, Snort uses a rule-based language to analyze packets and identify potential threats such as malware, port scans, or unauthorized access. It's widely used for network security due to its flexibility, efficiency, and active community support.

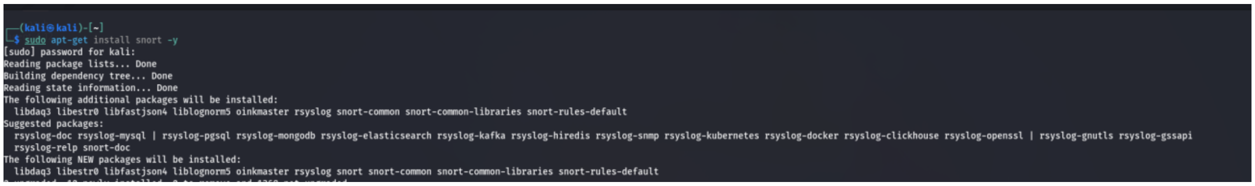
Wazuh with Snort!

Snort and Wazuh can be integrated to enhance network and host-based security monitoring. Snort detects and generates alerts for suspicious network traffic, while Wazuh collects, analyzes, and correlates these alerts with other system logs. By integrating both, security teams gain centralized visibility and improved threat detection, combining Snort's network-level insights with Wazuh's endpoint and log analysis capabilities.

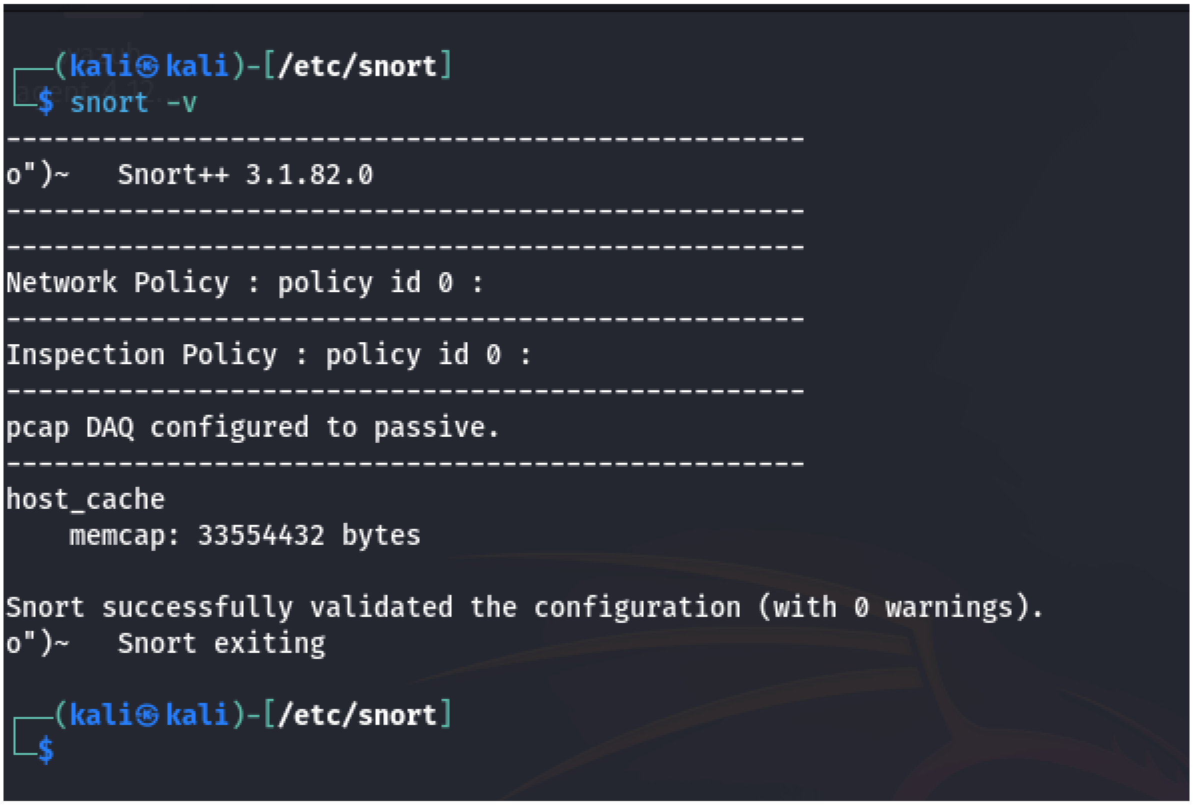
Snort Installation:

Command:

Sudo apt-get install snort –y



```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get install snort -y
[sudo] password for kali:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libdaq3 libestr0 libfastjson4 liblognorm5 oinkmaster rsyslog snort-common snort-common-libraries snort-rules-default
Suggested packages:
  rsyslog-doc rsyslog-mysql | rsyslog-pgsql rsyslog-mongodb rsyslog-elasticsearch rsyslog-kafka rsyslog-hiredis rsyslog-snmp rsyslog-kubernetes rsyslog-docker rsyslog-clickhouse rsyslog-openssl | rsyslog-gnutls rsyslog-gssapi
  rsyslog-relp snort-doc
The following NEW packages will be installed:
  libdaq3 libestr0 libfastjson4 liblognorm5 oinkmaster rsyslog snort snort-common snort-common-libraries snort-rules-default
```

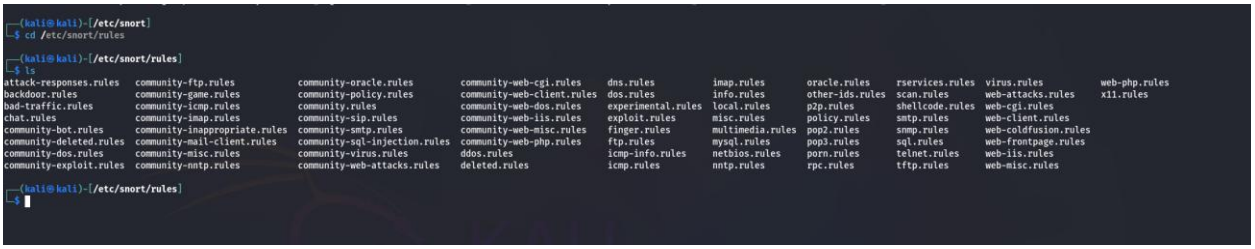


```
┌──(kali㉿kali)-[/etc/snort]
└─$ snort -v
--------------------------------------------------
o")~   Snort++ 3.1.82.0
--------------------------------------------------
--------------------------------------------------
Network Policy : policy id 0 :
--------------------------------------------------
Inspection Policy : policy id 0 :
--------------------------------------------------
pcap DAQ configured to passive.
--------------------------------------------------
host_cache
    memcap: 33554432 bytes

Snort successfully validated the configuration (with 0 warnings).
o")~   Snort exiting

┌──(kali㉿kali)-[/etc/snort]
└─$
```

now change the directory to /etc/snort/rules

command:

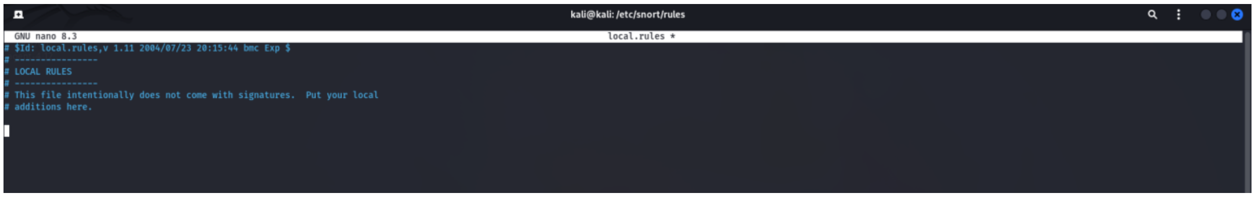
/etc/snort/rules/local.rules



```
┌──(kali㉿kali)-[/etc/snort]
└─$ cd /etc/snort/rules

┌──(kali㉿kali)-[/etc/snort/rules]
└─$ ls
attack-responses.rules   community-ftp.rules            community-oracle.rules         community-web-cgi.rules     dns.rules           imap.rules        oracle.rules     rservices.rules  virus.rules           web-php.rules
backdoor.rules           community-game.rules           community-policy.rules         community-web-client.rules  dos.rules           info.rules        other-ids.rules  scan.rules       web-attacks.rules     x11.rules
bad-traffic.rules        community-icmp.rules           community.rules                community-web-dos.rules     experimental.rules  local.rules       p2p.rules        shellcode.rules  web-cgi.rules
chat.rules               community-imap.rules           community-sip.rules            community-web-iis.rules     exploit.rules       misc.rules        policy.rules     smtp.rules       web-client.rules
community-bot.rules      community-inappropriate.rules  community-smtp.rules           community-web-misc.rules    finger.rules        multimedia.rules  pop2.rules       snmp.rules       web-coldfusion.rules
community-deleted.rules  community-mail-client.rules    community-sql-injection.rules  community-web-php.rules     ftp.rules           mysql.rules       pop3.rules       sql.rules        web-frontpage.rules
community-dos.rules      community-misc.rules           community-virus.rules          ddos.rules                  icmp-info.rules     netbios.rules     porn.rules       telnet.rules     web-iis.rules
community-exploit.rules  community-nntp.rules           community-web-attacks.rules    deleted.rules               icmp.rules          nntp.rules        rpc.rules        tftp.rules       web-misc.rules

┌──(kali㉿kali)-[/etc/snort/rules]
└─$ 
```

Open the local.rules file

Nano local.rules



```
  GNU nano 8.3                                      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.
```

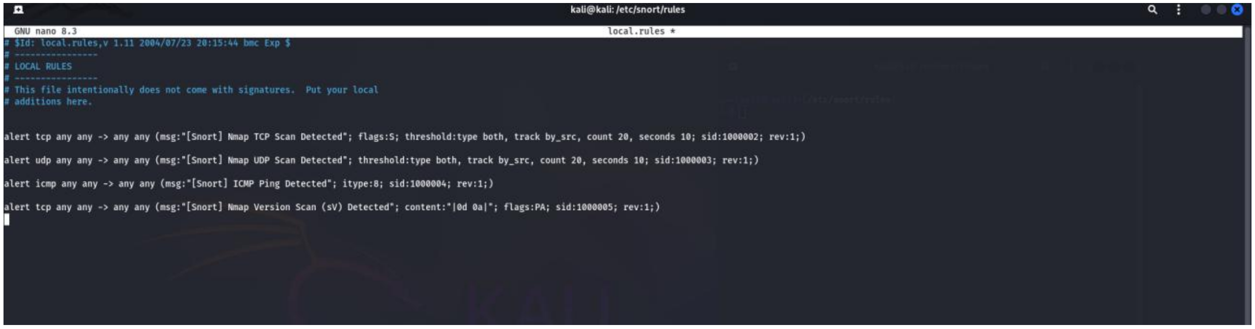
Now add these rules in the file

alert tcp any any -> any any (msg:"[Snort] Nmap TCP Scan Detected"; flags:S; sid:1000002; rev:2;)

alert udp any any -> any any (msg:"[Snort] Nmap UDP Scan Detected"; sid:1000003; rev:2;)

alert icmp any any -> any any (msg:"[Snort] ICMP Ping Detected"; itype:8; sid:1000004; rev:2;)

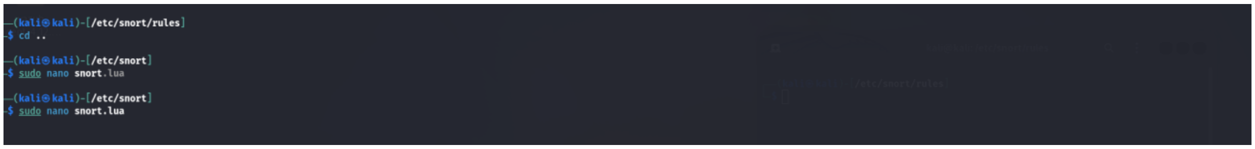
alert tcp any any -> any any (msg:"[Snort] Nmap Version Scan (sV) Detected"; content:"|0d 0a|"; flags:PA; sid:1000005; rev:2;)



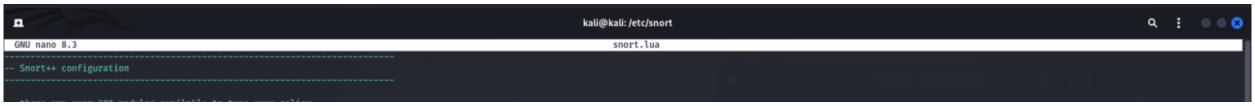
```
  GNU nano 8.3                                      local.rules *
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# ----------------
# LOCAL RULES
# ----------------
# This file intentionally does not come with signatures.  Put your local
# additions here.

alert tcp any any -> any any (msg:"[Snort] Nmap TCP Scan Detected"; flags:S; threshold:type both, track by_src, count 20, seconds 10; sid:1000002; rev:1;)

alert udp any any -> any any (msg:"[Snort] Nmap UDP Scan Detected"; threshold:type both, track by_src, count 20, seconds 10; sid:1000003; rev:1;)

alert icmp any any -> any any (msg:"[Snort] ICMP Ping Detected"; itype:8; sid:1000004; rev:1;)

alert tcp any any -> any any (msg:"[Snort] Nmap Version Scan (sV) Detected"; content:"|0d 0a|"; flags:PA; sid:1000005; rev:1;)
```

Now save the file and change the directory to snort.lua file.

i.e /etc/snort

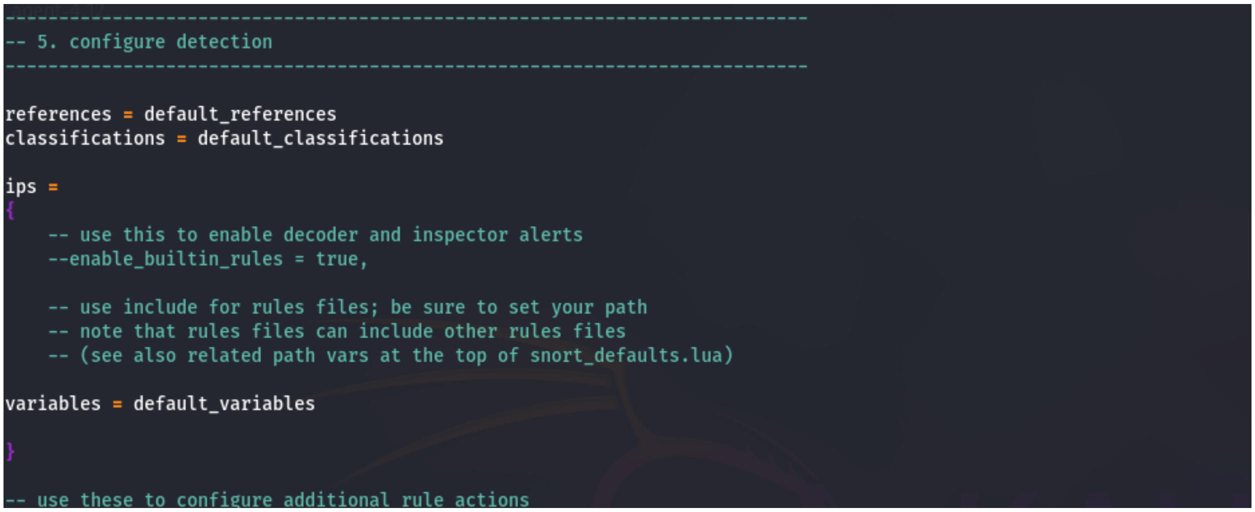


```
┌──(kali㉿kali)-[/etc/snort/rules]
└─$ cd ..

┌──(kali㉿kali)-[/etc/snort]
└─$ sudo nano snort.lua

┌──(kali㉿kali)-[/etc/snort]
└─$ sudo nano snort.lua
```



```
  GNU nano 8.3                                      snort.lua
---------------------------------------------------------------------------
-- Snort++ configuration
---------------------------------------------------------------------------
```

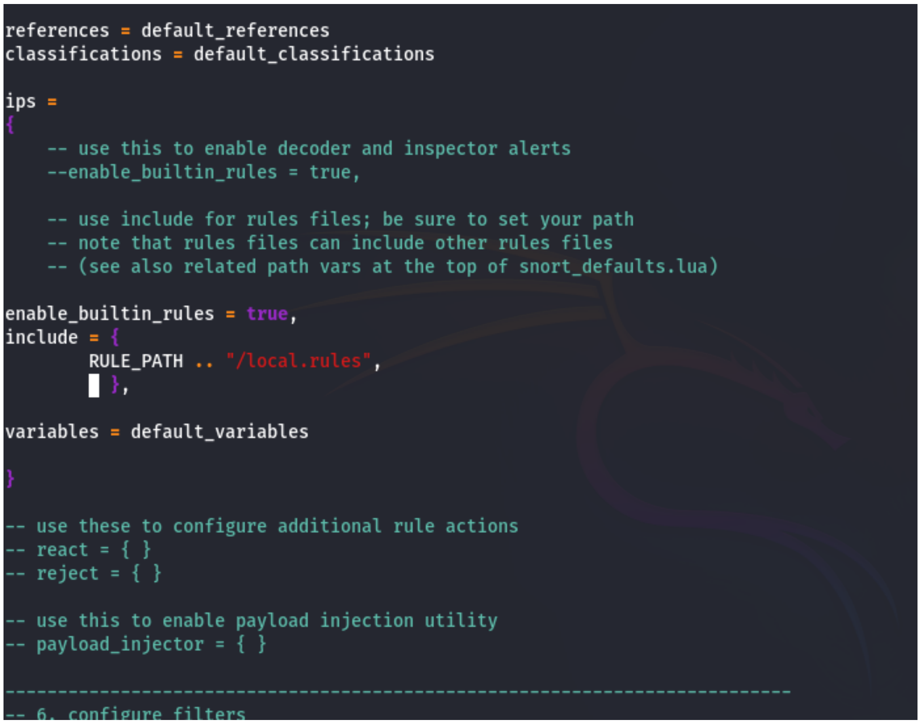
After opening the **snort.lua** look for configure detection section



```
---------------------------------------------------------------------------
-- 5. configure detection
---------------------------------------------------------------------------

references = default_references
classifications = default_classifications

ips =
{
    -- use this to enable decoder and inspector alerts
    --enable_builtin_rules = true,

    -- use include for rules files; be sure to set your path
    -- note that rules files can include other rules files
    -- (see also related path vars at the top of snort_defaults.lua)

variables = default_variables

}

-- use these to configure additional rule actions
```

Now add this code in configure detection section.

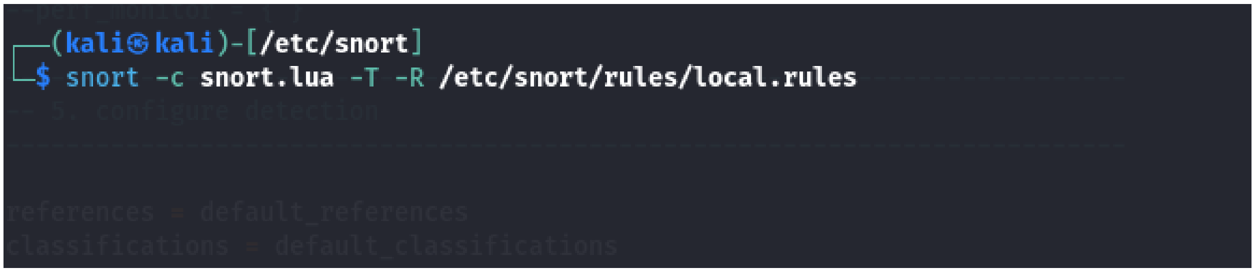
```
enable_builtin_rules = true,

include = {

      RULE_PATH .. "/local.rules",

   },
```



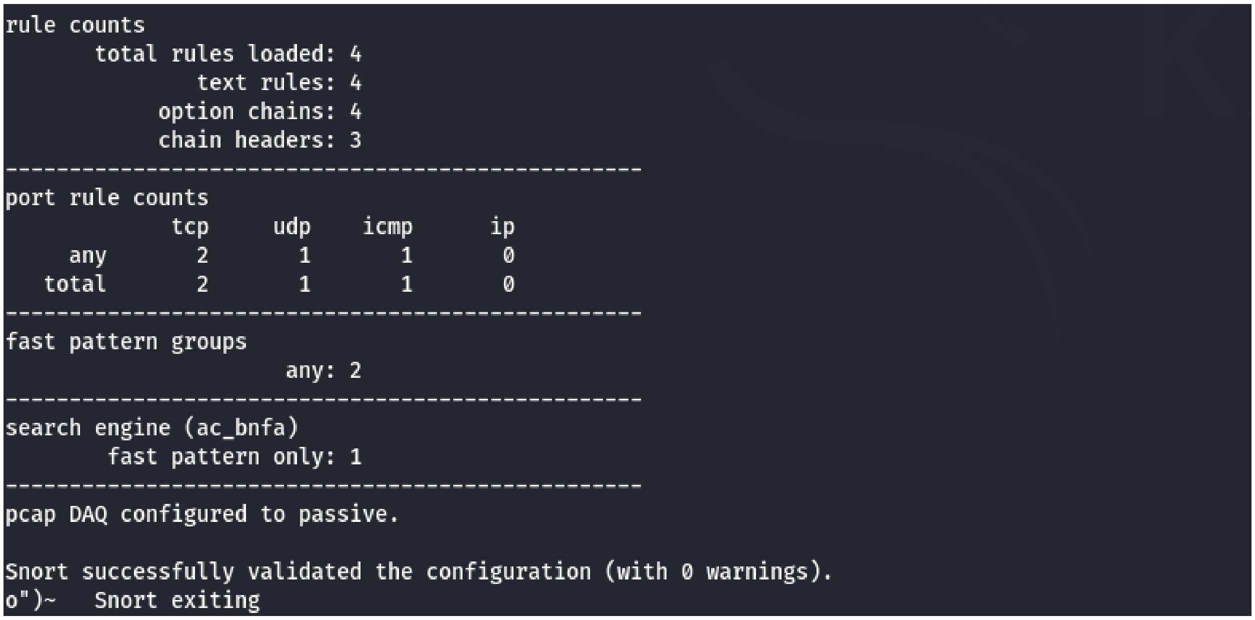
```
references = default_references
classifications = default_classifications

ips =
{
    -- use this to enable decoder and inspector alerts
    --enable_builtin_rules = true,

    -- use include for rules files; be sure to set your path
    -- note that rules files can include other rules files
    -- (see also related path vars at the top of snort_defaults.lua)

enable_builtin_rules = true,
include = {
        RULE_PATH .. "/local.rules",
        },

variables = default_variables

}

-- use these to configure additional rule actions
-- react = { }
-- reject = { }

-- use this to enable payload injection utility
-- payload_injector = { }

---------------------------------------------------------------------------
-- 6. configure filters
```

Now save and test the file.

Note: I am adding again –R because I am facing rule path error



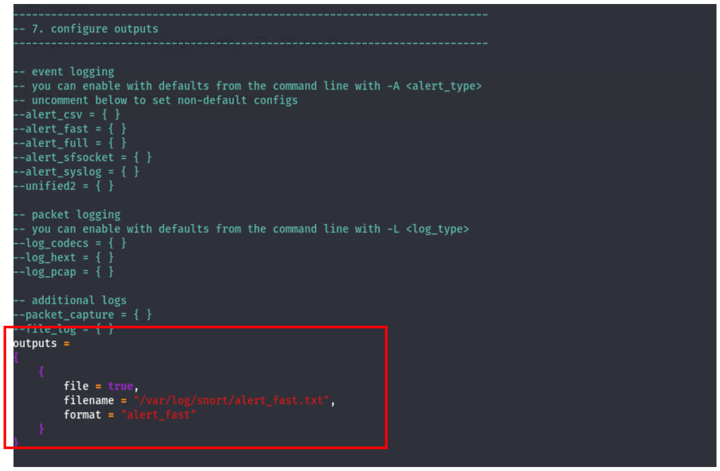
```
┌──(kali㉿kali)-[/etc/snort]
└─$ snort -c snort.lua -T -R /etc/snort/rules/local.rules
```



```
rule counts
       total rules loaded: 4
               text rules: 4
            option chains: 4
            chain headers: 3
--------------------------------------------------
port rule counts
             tcp     udp    icmp      ip
     any       2       1       1       0
   total       2       1       1       0
--------------------------------------------------
fast pattern groups
                      any: 2
--------------------------------------------------
search engine (ac_bnfa)
        fast pattern only: 1
--------------------------------------------------
pcap DAQ configured to passive.

Snort successfully validated the configuration (with 0 warnings).
o")~   Snort exiting
```

Now again open the snort.lua file to configure output

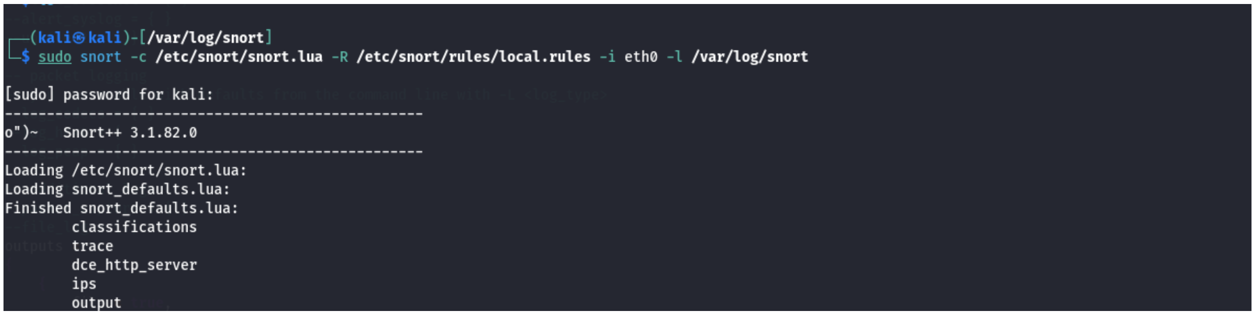
Add this piece of code in the output configuration and save the file.



```
---------------------------------------------------------------------------
-- 7. configure outputs
---------------------------------------------------------------------------

-- event logging
-- you can enable with defaults from the command line with -A <alert_type>
-- uncomment below to set non-default configs
--alert_csv = { }
--alert_fast = { }
--alert_full = { }
--alert_sfsocket = { }
--alert_syslog = { }
--unified2 = { }

-- packet logging
-- you can enable with defaults from the command line with -L <log_type>
--log_codecs = { }
--log_hext = { }
--log_pcap = { }

-- additional logs
--packet_capture = { }
--file_log = { }
outputs =
{
    {
        file = true,
        filename = "/var/log/snort/alert_fast.txt",
        format = "alert_fast"
    }
}
```

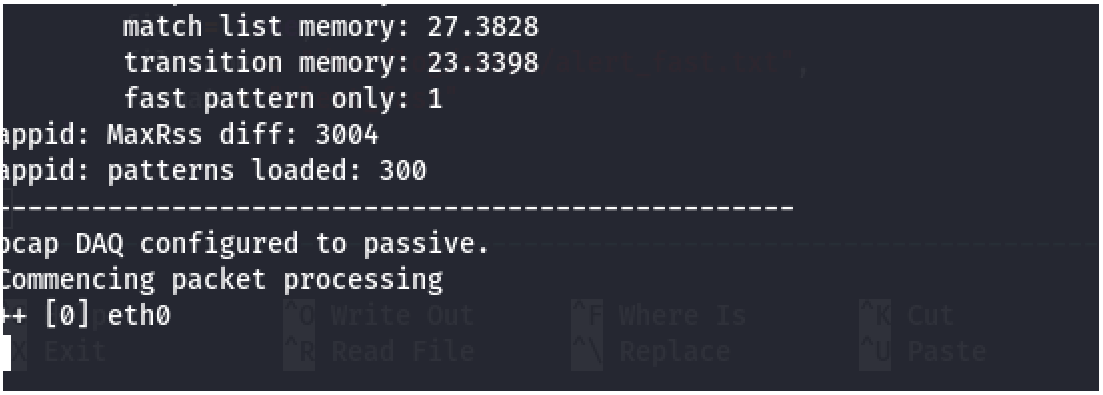
After writing again check the working by running this command

sudo snort -c /etc/snort/snort.lua -R /etc/snort/rules/local.rules -i eth0 -l /var/log/snort



```
┌──(kali㉿kali)-[/var/log/snort]
└─$ sudo snort -c /etc/snort/snort.lua -R /etc/snort/rules/local.rules -i eth0 -l /var/log/snort
[sudo] password for kali:
--------------------------------------------------
o")~   Snort++ 3.1.82.0
--------------------------------------------------
Loading /etc/snort/snort.lua:
Loading snort_defaults.lua:
Finished snort_defaults.lua:
        classifications
        trace
        dce_http_server
        ips
        output
```

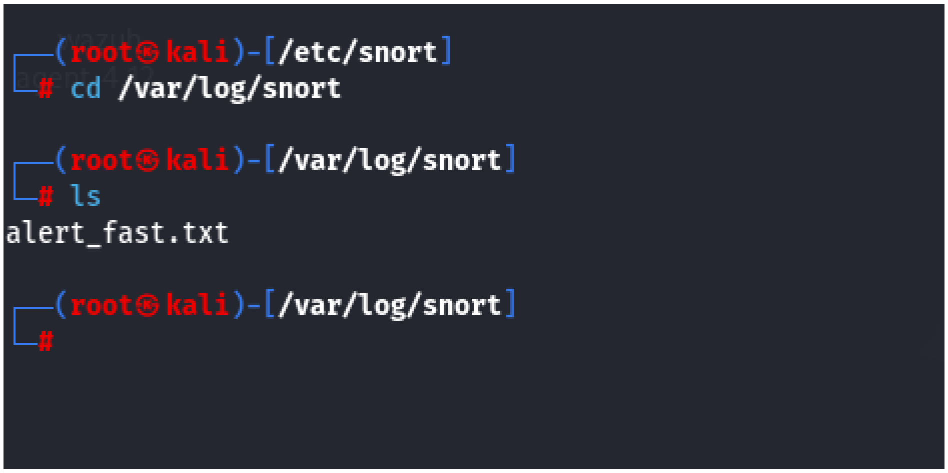
It's configured and running properly



```
        match list memory: 27.3828
        transition memory: 23.3398
        fast pattern only: 1
appid: MaxRss diff: 3004
appid: patterns loaded: 300
--------------------------------------------------
pcap DAQ configured to passive.
Commencing packet processing
++ [0] eth0
```

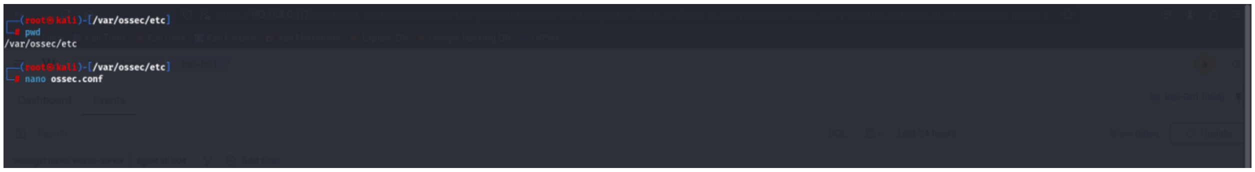
Now check the file **fast_alert** is created or not

To check go to **/var/log/snort folder**



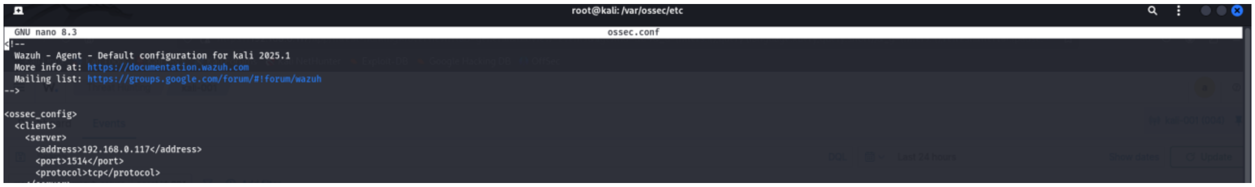
```
┌──(root㉿kali)-[/etc/snort]
└─# cd /var/log/snort

┌──(root㉿kali)-[/var/log/snort]
└─# ls
alert_fast.txt

┌──(root㉿kali)-[/var/log/snort]
└─#
```

After that open **Ossec.conf** Wazuh agent file to read the output.



```
┌──(root㉿kali)-[/var/ossec/etc]
└─# pwd
/var/ossec/etc

┌──(root㉿kali)-[/var/ossec/etc]
└─# nano ossec.conf
```

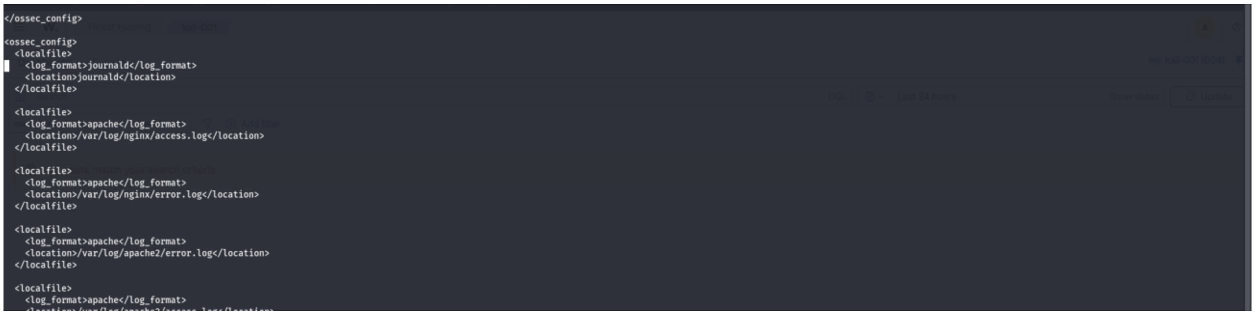
Nano ossec.conf



root@kali: /var/ossec/etc

```
  GNU nano 8.3                                   ossec.conf
<!--
  Wazuh - Agent - Default configuration for kali 2025.1
  More info at: https://documentation.wazuh.com
  Mailing list: https://groups.google.com/forum/#!forum/wazuh
-->

<ossec_config>
  <client>
    <server>
      <address>192.168.0.117</address>
      <port>1514</port>
      <protocol>tcp</protocol>
```

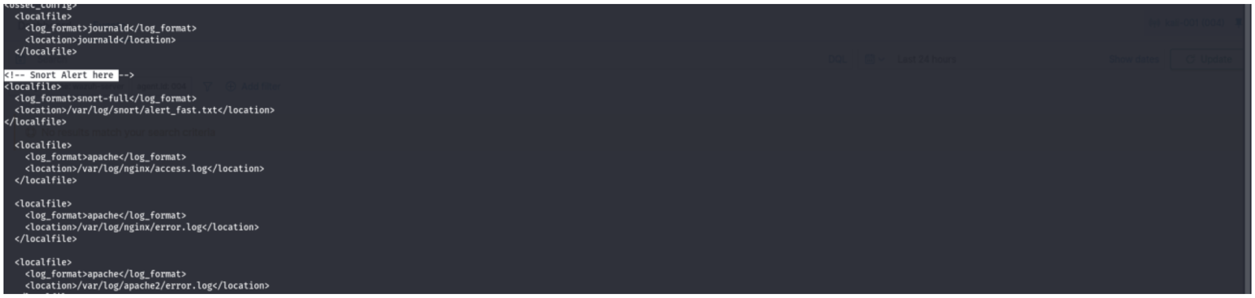
Now goto localfile tags section



```
</ossec_config>

<ossec_config>
  <localfile>
    <log_format>journald</log_format>
    <location>journald</location>
  </localfile>

  <localfile>
    <log_format>apache</log_format>
    <location>/var/log/nginx/access.log</location>
  </localfile>

  <localfile>
    <log_format>apache</log_format>
    <location>/var/log/nginx/error.log</location>
  </localfile>

  <localfile>
    <log_format>apache</log_format>
    <location>/var/log/apache2/error.log</location>
  </localfile>

  <localfile>
    <log_format>apache</log_format>
    <location>/var/log/apache2/access.log</location>
```

Paste this code in the localfile path

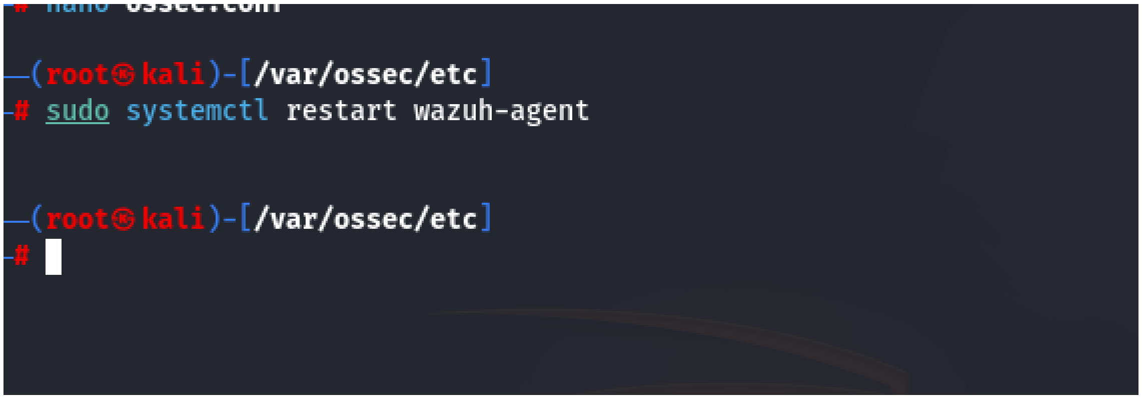
```
<localfile>
  <log_format>snort-full</log_format>
  <location>/var/log/snort/alert_fast.txt</location>
</localfile>
```



```
<ossec_config>
  <localfile>
    <log_format>journald</log_format>
    <location>journald</location>
  </localfile>

<!-- Snort Alert here -->
<localfile>
  <log_format>snort-full</log_format>
  <location>/var/log/snort/alert_fast.txt</location>
</localfile>

  <localfile>
    <log_format>apache</log_format>
    <location>/var/log/nginx/access.log</location>
  </localfile>

  <localfile>
    <log_format>apache</log_format>
    <location>/var/log/nginx/error.log</location>
  </localfile>

  <localfile>
    <log_format>apache</log_format>
    <location>/var/log/apache2/error.log</location>
```

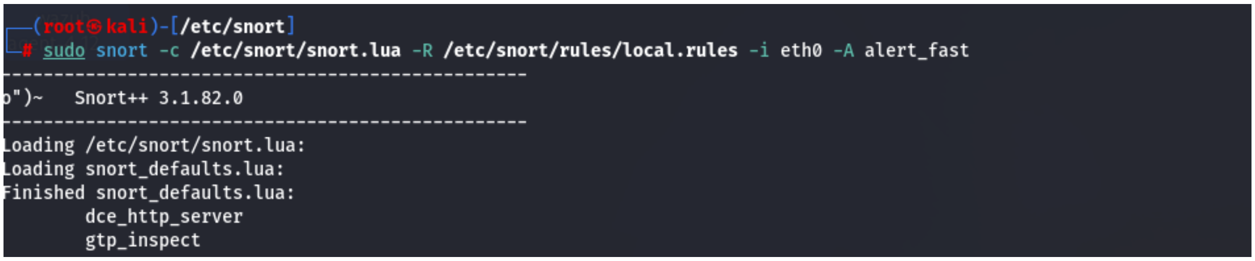
Now save the file and reastart the Wazuh-agent

```
sudo systemctl restart wazuh-agent
```



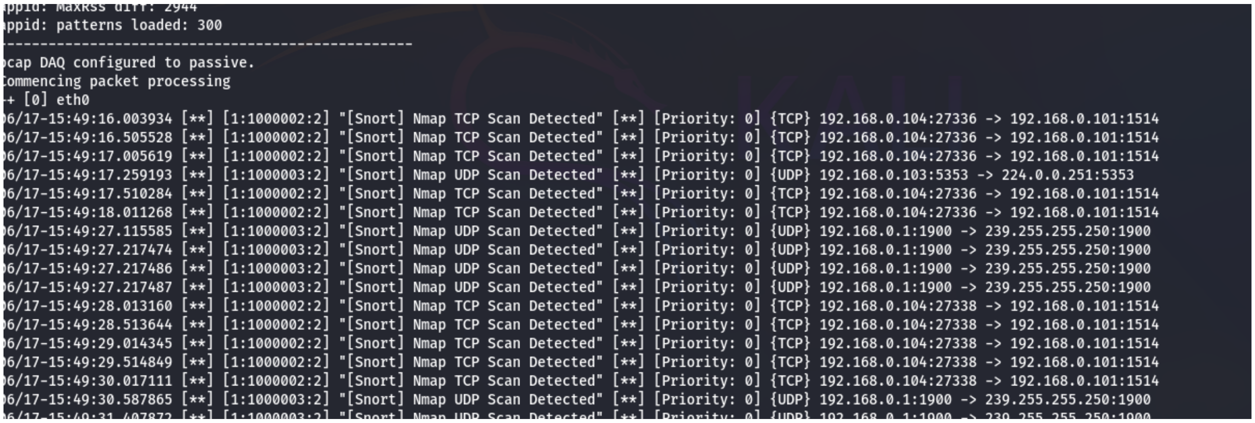
```
# nano ossec.conf

┌──(root㉿kali)-[/var/ossec/etc]
└─# sudo systemctl restart wazuh-agent

┌──(root㉿kali)-[/var/ossec/etc]
└─# 
```

Now run snort in ids mode



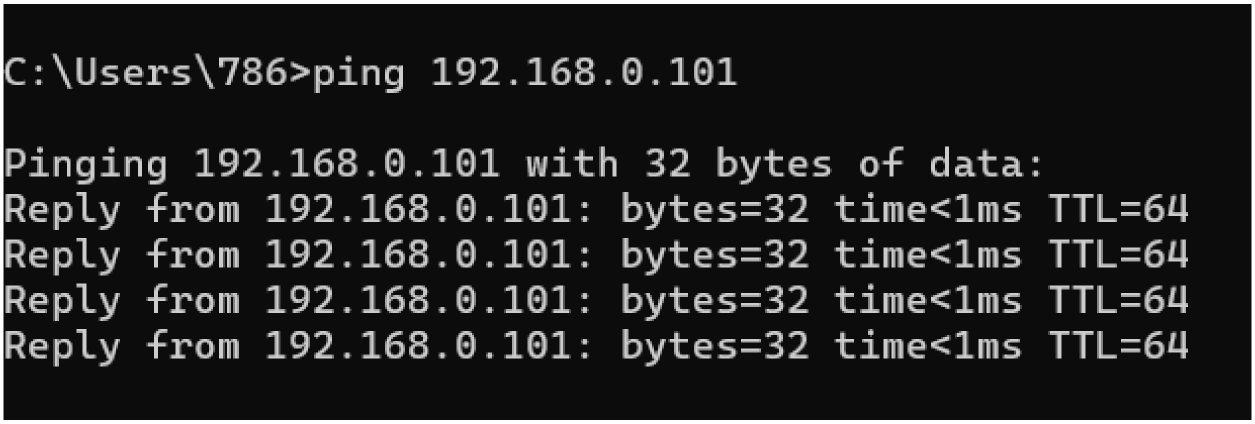
```
┌──(root㉿kali)-[/etc/snort]
└─# sudo snort -c /etc/snort/snort.lua -R /etc/snort/rules/local.rules -i eth0 -A alert_fast
--------------------------------------------------
o")~   Snort++ 3.1.82.0
--------------------------------------------------
Loading /etc/snort/snort.lua:
Loading snort_defaults.lua:
Finished snort_defaults.lua:
        dce_http_server
        gtp_inspect
```

Now scan the ip with nmap and it will generate alert on the console.



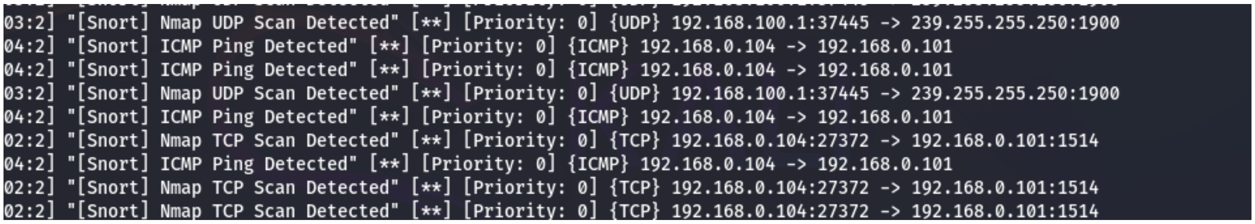
```
ppid: MaxRss diff: 2944
ppid: patterns loaded: 300
--------------------------------------------------
pcap DAQ configured to passive.
Commencing packet processing
++ [0] eth0
06/17-15:49:16.003934 [**] [1:1000002:2] "[Snort] Nmap TCP Scan Detected" [**] [Priority: 0] {TCP} 192.168.0.104:27336 -> 192.168.0.101:1514
06/17-15:49:16.505528 [**] [1:1000002:2] "[Snort] Nmap TCP Scan Detected" [**] [Priority: 0] {TCP} 192.168.0.104:27336 -> 192.168.0.101:1514
06/17-15:49:17.005619 [**] [1:1000002:2] "[Snort] Nmap TCP Scan Detected" [**] [Priority: 0] {TCP} 192.168.0.104:27336 -> 192.168.0.101:1514
06/17-15:49:17.259193 [**] [1:1000003:2] "[Snort] Nmap UDP Scan Detected" [**] [Priority: 0] {UDP} 192.168.0.103:5353 -> 224.0.0.251:5353
06/17-15:49:17.510284 [**] [1:1000002:2] "[Snort] Nmap TCP Scan Detected" [**] [Priority: 0] {TCP} 192.168.0.104:27336 -> 192.168.0.101:1514
06/17-15:49:18.011268 [**] [1:1000002:2] "[Snort] Nmap TCP Scan Detected" [**] [Priority: 0] {TCP} 192.168.0.104:27336 -> 192.168.0.101:1514
06/17-15:49:27.115585 [**] [1:1000003:2] "[Snort] Nmap UDP Scan Detected" [**] [Priority: 0] {UDP} 192.168.0.1:1900 -> 239.255.255.250:1900
06/17-15:49:27.217474 [**] [1:1000003:2] "[Snort] Nmap UDP Scan Detected" [**] [Priority: 0] {UDP} 192.168.0.1:1900 -> 239.255.255.250:1900
06/17-15:49:27.217486 [**] [1:1000003:2] "[Snort] Nmap UDP Scan Detected" [**] [Priority: 0] {UDP} 192.168.0.1:1900 -> 239.255.255.250:1900
06/17-15:49:27.217487 [**] [1:1000003:2] "[Snort] Nmap UDP Scan Detected" [**] [Priority: 0] {UDP} 192.168.0.1:1900 -> 239.255.255.250:1900
06/17-15:49:28.013160 [**] [1:1000002:2] "[Snort] Nmap TCP Scan Detected" [**] [Priority: 0] {TCP} 192.168.0.104:27338 -> 192.168.0.101:1514
06/17-15:49:28.513644 [**] [1:1000002:2] "[Snort] Nmap TCP Scan Detected" [**] [Priority: 0] {TCP} 192.168.0.104:27338 -> 192.168.0.101:1514
06/17-15:49:29.014345 [**] [1:1000002:2] "[Snort] Nmap TCP Scan Detected" [**] [Priority: 0] {TCP} 192.168.0.104:27338 -> 192.168.0.101:1514
06/17-15:49:29.514849 [**] [1:1000002:2] "[Snort] Nmap TCP Scan Detected" [**] [Priority: 0] {TCP} 192.168.0.104:27338 -> 192.168.0.101:1514
06/17-15:49:30.017111 [**] [1:1000002:2] "[Snort] Nmap TCP Scan Detected" [**] [Priority: 0] {TCP} 192.168.0.104:27338 -> 192.168.0.101:1514
06/17-15:49:30.587865 [**] [1:1000003:2] "[Snort] Nmap UDP Scan Detected" [**] [Priority: 0] {UDP} 192.168.0.1:1900 -> 239.255.255.250:1900
06/17-15:49:31.407872 [**] [1:1000003:2] "[Snort] Nmap UDP Scan Detected" [**] [Priority: 0] {UDP} 192.168.0.1:1900 -> 239.255.255.250:1900
```

Ping the ip



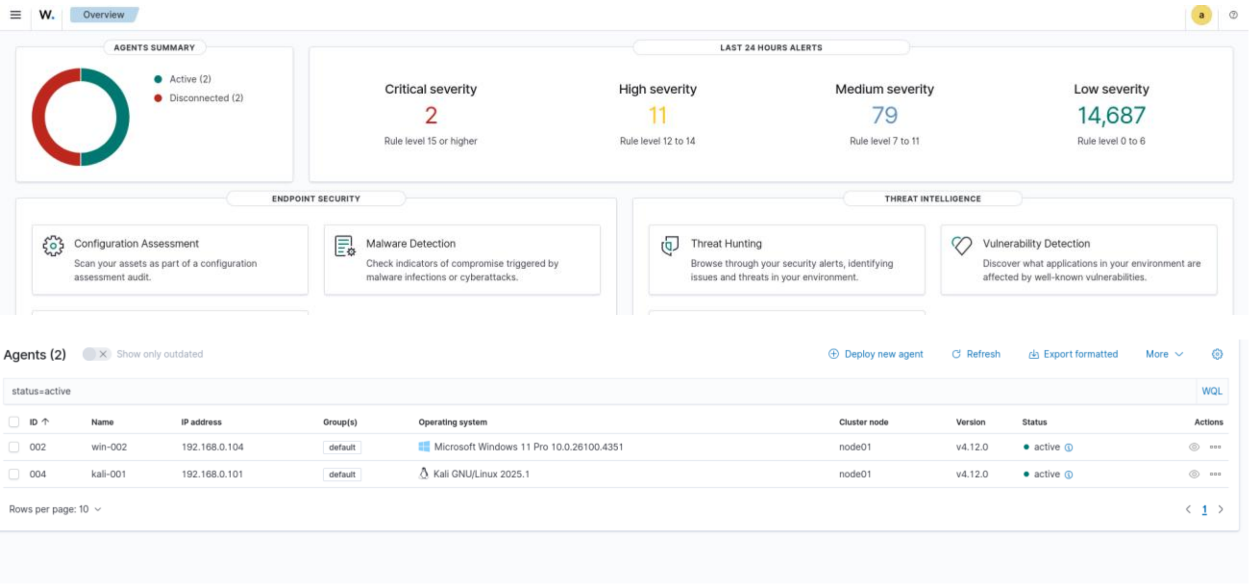
```
C:\Users\786>ping 192.168.0.101

Pinging 192.168.0.101 with 32 bytes of data:
Reply from 192.168.0.101: bytes=32 time<1ms TTL=64
Reply from 192.168.0.101: bytes=32 time<1ms TTL=64
Reply from 192.168.0.101: bytes=32 time<1ms TTL=64
Reply from 192.168.0.101: bytes=32 time<1ms TTL=64
```

Output:



```
03:2] "[Snort] Nmap UDP Scan Detected" [**] [Priority: 0] {UDP} 192.168.100.1:37445 -> 239.255.255.250:1900
04:2] "[Snort] ICMP Ping Detected" [**] [Priority: 0] {ICMP} 192.168.0.104 -> 192.168.0.101
04:2] "[Snort] ICMP Ping Detected" [**] [Priority: 0] {ICMP} 192.168.0.104 -> 192.168.0.101
03:2] "[Snort] Nmap UDP Scan Detected" [**] [Priority: 0] {UDP} 192.168.100.1:37445 -> 239.255.255.250:1900
04:2] "[Snort] ICMP Ping Detected" [**] [Priority: 0] {ICMP} 192.168.0.104 -> 192.168.0.101
02:2] "[Snort] Nmap TCP Scan Detected" [**] [Priority: 0] {TCP} 192.168.0.104:27372 -> 192.168.0.101:1514
04:2] "[Snort] ICMP Ping Detected" [**] [Priority: 0] {ICMP} 192.168.0.104 -> 192.168.0.101
02:2] "[Snort] Nmap TCP Scan Detected" [**] [Priority: 0] {TCP} 192.168.0.104:27372 -> 192.168.0.101:1514
02:2] "[Snort] Nmap TCP Scan Detected" [**] [Priority: 0] {TCP} 192.168.0.104:27372 -> 192.168.0.101:1514
```

Now open Wazuh-dashboard:



AGENTS SUMMARY

- ● Active (2)
- ● Disconnected (2)

LAST 24 HOURS ALERTS

Critical severity	High severity	Medium severity	Low severity
2	11	79	14,687
Rule level 15 or higher	Rule level 12 to 14	Rule level 7 to 11	Rule level 0 to 6

ENDPOINT SECURITY

Configuration Assessment
Scan your assets as part of a configuration assessment audit.

Malware Detection
Check indicators of compromise triggered by malware infections or cyberattacks.

THREAT INTELLIGENCE

Threat Hunting
Browse through your security alerts, identifying issues and threats in your environment.

Vulnerability Detection
Discover what applications in your environment are affected by well-known vulnerabilities.

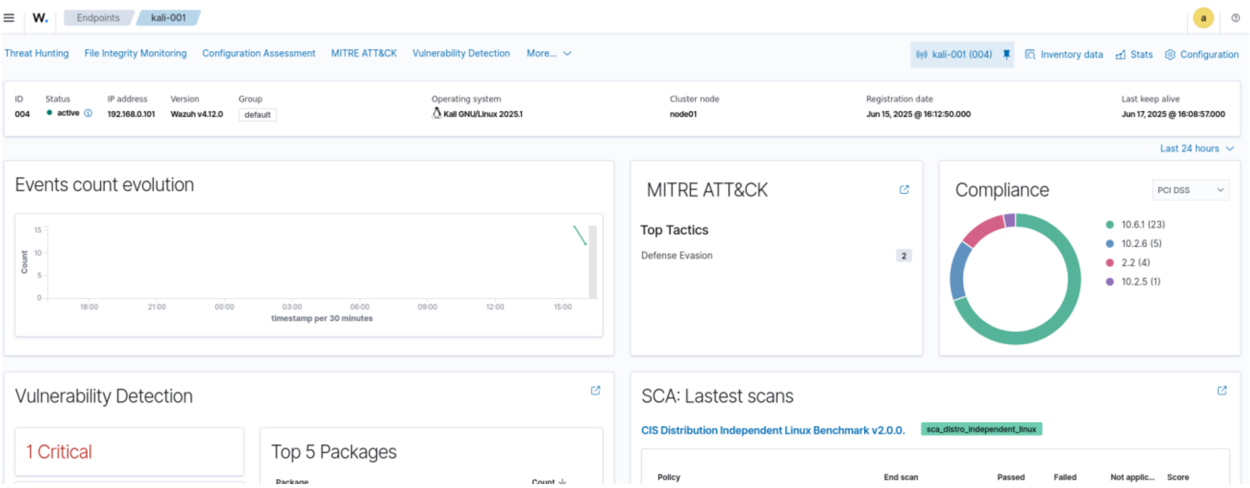
Agents (2) Show only outdated

status=active

ID ↑	Name	IP address	Group(s)	Operating system	Cluster node	Version	Status	Actions
002	win-002	192.168.0.104	default	Microsoft Windows 11 Pro 10.0.26100.4351	node01	v4.12.0	● active	
004	kali-001	192.168.0.101	default	Kali GNU/Linux 2025.1	node01	v4.12.0	● active	

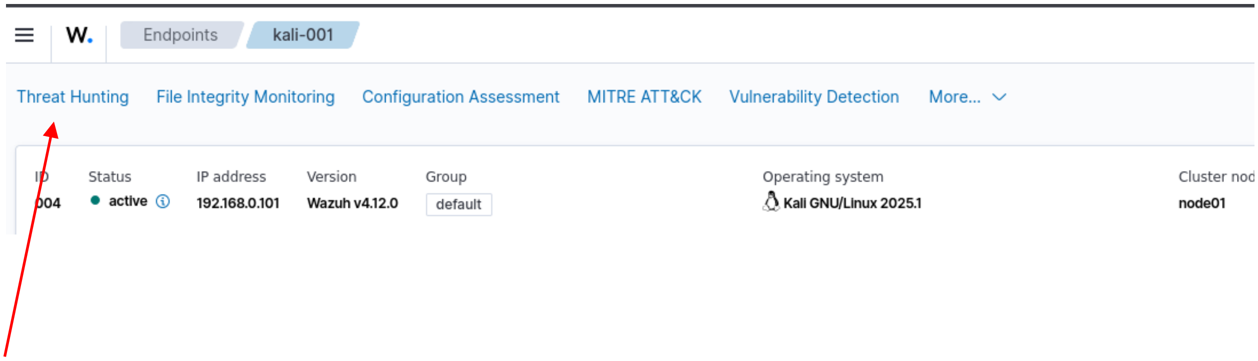
Rows per page: 10

Open the kali/Ubuntu agent:

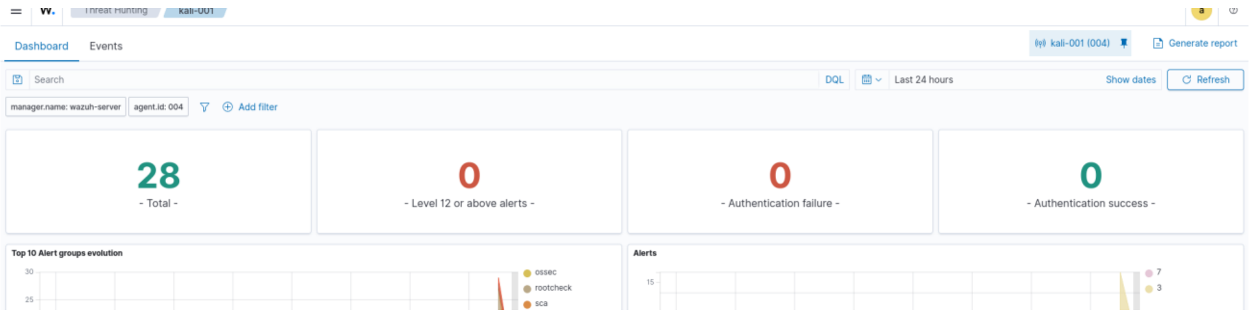


ID	Status	IP address	Version	Group	Operating system	Cluster node	Registration date	Last keep alive
004	● active	192.168.0.101	Wazuh v4.12.0	default	Kali GNU/Linux 2025.1	node01	Jun 15, 2025 @ 16:12:50.000	Jun 17, 2025 @ 16:08:57.000

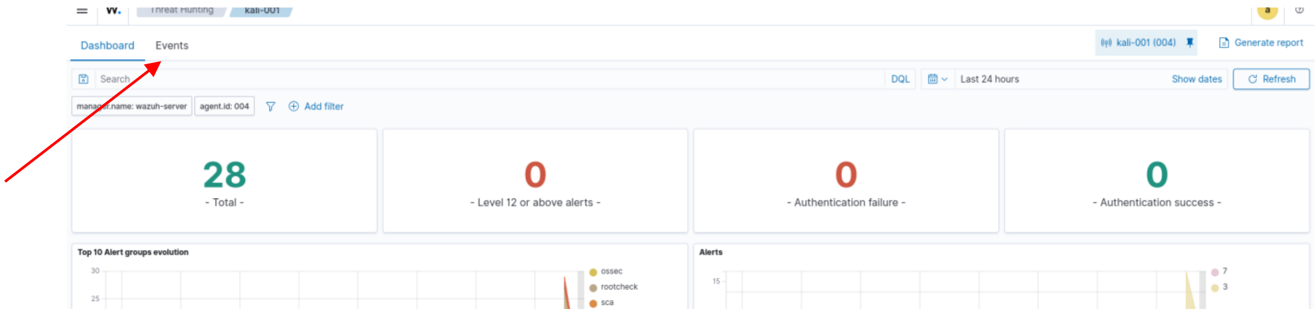
Now click on threat hunting



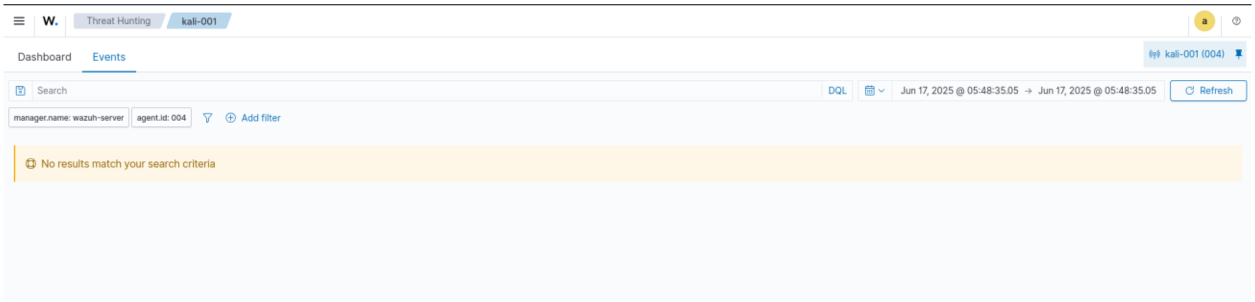
ID	Status	IP address	Version	Group	Operating system	Cluster nod
004	● active	192.168.0.101	Wazuh v4.12.0	default	Kali GNU/Linux 2025.1	node01



Now click on Events:



After opening the events tab it's not showing logs



No results match your search criteria

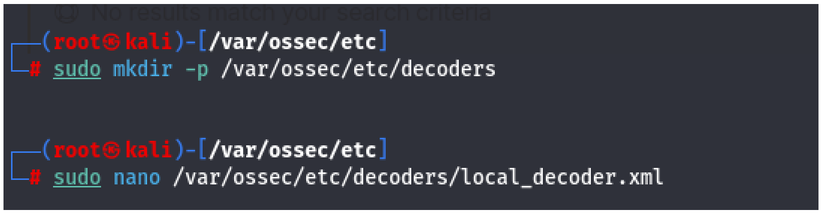
To solve this problem we need to define decoder. Open decoder folder

Cd /var/ossec/decoders

If not found create it using: **sudo mkdir -p /var/ossec/etc/decoders**

And after that create custom decoder file for snort rules.

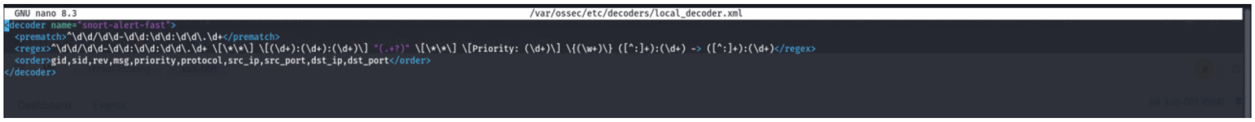
sudo nano /var/ossec/etc/decoders/local_decoder.xml



```
┌──(root㉿kali)-[/var/ossec/etc]
└─# sudo mkdir -p /var/ossec/etc/decoders


┌──(root㉿kali)-[/var/ossec/etc]
└─# sudo nano /var/ossec/etc/decoders/local_decoder.xml
```

After creating the file add this decoding code:

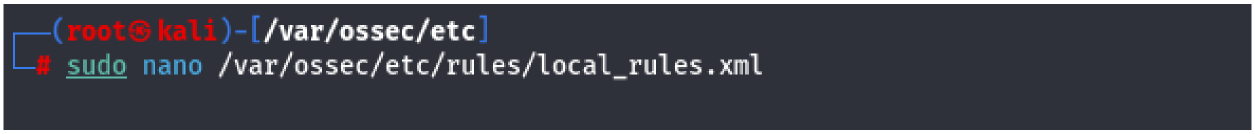
<decoder name="snort-alert-fast">
<prematch>^\d\d/\d\d-\d\d:\d\d:\d\d\.\d+</prematch>
<regex>^\d\d/\d\d-\d\d:\d\d:\d\d\.\d+ \[**\] \[(\d+):(\d+):(\d+)\] "(.+?)" \[**\] \[Priority: (\d+)\] \{(\w+)\} ([^:]+):(\d+) -> ([^:]+):(\d+)</regex>
<order>gid,sid,rev,msg,priority,protocol,src_ip,src_port,dst_ip,dst_port</order>
</decoder>



```
GNU nano 8.3                                   /var/ossec/etc/decoders/local_decoder.xml
<decoder name="snort-alert-fast">
  <prematch>^\d\d/\d\d-\d\d:\d\d:\d\d\.\d+</prematch>
  <regex>^\d\d/\d\d-\d\d:\d\d:\d\d\.\d+ \[\*\*\] \[(\d+):(\d+):(\d+)\] "(.+?)" \[\*\*\] \[Priority: (\d+)\] \{(\w+)\} ([^:]+):(\d+) -> ([^:]+):(\d+)</regex>
  <order>gid,sid,rev,msg,priority,protocol,src_ip,src_port,dst_ip,dst_port</order>
</decoder>
```

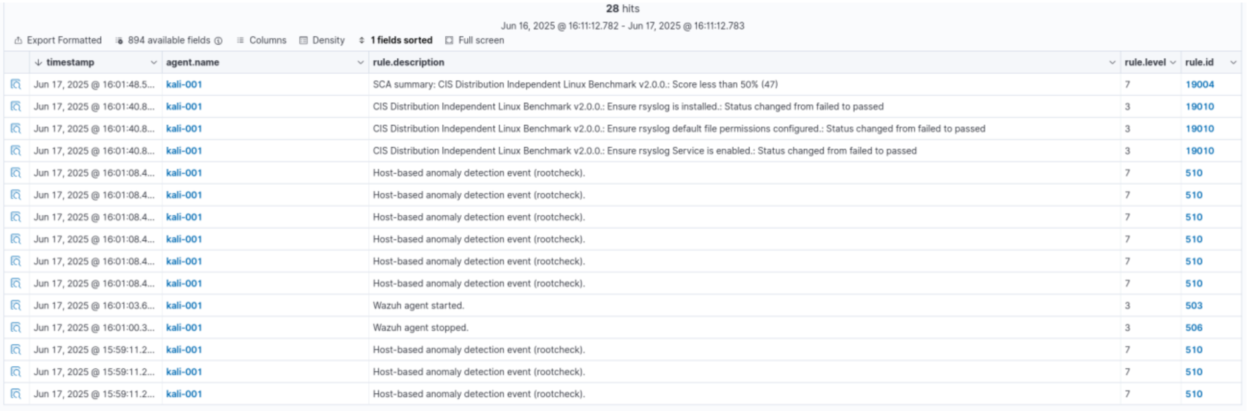
Now go to **local.rule** file

sudo nano /var/ossec/etc/rules/local_rules.xml



```
┌──(root㉿kali)-[/var/ossec/etc]
└─# sudo nano /var/ossec/etc/rules/local_rules.xml
```

Here’s the IDS logs:



timestamp	agent.name	rule.description	rule.level	rule.id
Jun 17, 2025 @ 16:01:48.5...	kali-001	SCA summary: CIS Distribution Independent Linux Benchmark v2.0.0.: Score less than 50% (47)	7	19004
Jun 17, 2025 @ 16:01:40.8...	kali-001	CIS Distribution Independent Linux Benchmark v2.0.0.: Ensure rsyslog is installed.: Status changed from failed to passed	3	19010
Jun 17, 2025 @ 16:01:40.8...	kali-001	CIS Distribution Independent Linux Benchmark v2.0.0.: Ensure rsyslog default file permissions configured.: Status changed from failed to passed	3	19010
Jun 17, 2025 @ 16:01:40.8...	kali-001	CIS Distribution Independent Linux Benchmark v2.0.0.: Ensure rsyslog Service is enabled.: Status changed from failed to passed	3	19010
Jun 17, 2025 @ 16:01:08.4...	kali-001	Host-based anomaly detection event (rootcheck).	7	510
Jun 17, 2025 @ 16:01:08.4...	kali-001	Host-based anomaly detection event (rootcheck).	7	510
Jun 17, 2025 @ 16:01:08.4...	kali-001	Host-based anomaly detection event (rootcheck).	7	510
Jun 17, 2025 @ 16:01:08.4...	kali-001	Host-based anomaly detection event (rootcheck).	7	510
Jun 17, 2025 @ 16:01:08.4...	kali-001	Host-based anomaly detection event (rootcheck).	7	510
Jun 17, 2025 @ 16:01:08.4...	kali-001	Host-based anomaly detection event (rootcheck).	7	510
Jun 17, 2025 @ 16:01:03.6...	kali-001	Wazuh agent started.	3	503
Jun 17, 2025 @ 16:01:00.3...	kali-001	Wazuh agent stopped.	3	506
Jun 17, 2025 @ 15:59:11.2...	kali-001	Host-based anomaly detection event (rootcheck).	7	510
Jun 17, 2025 @ 15:59:11.2...	kali-001	Host-based anomaly detection event (rootcheck).	7	510
Jun 17, 2025 @ 15:59:11.2...	kali-001	Host-based anomaly detection event (rootcheck).	7	510

Summary:

Snort and Wazuh can be integrated to enhance network and host-based security monitoring. Snort detects and generates alerts for suspicious network traffic, while Wazuh collects, analyzes, and correlates these alerts with other system logs. By integrating both, security teams gain centralized visibility and improved threat detection, combining Snort's network-level insights with Wazuh's endpoint and log analysis capabilities.

Need training on Wazuh ?

Contact number: +923355345678

Email: sameeerishassan@gmail.com

LinkedIn: https://pk.linkedin.com/in/sameer-hassan-15a428255

Other SIEM

1. IBM Qradar

2. Splunk

3. Azure Sentinel