Lab 3: Endpoint security (EDR/SIEM)

Name: Mohamad Nour Shahin, Yehia Sobeh, Ali Hamdan, Matvey Makhnov

Group number: B22-CBS-01

You may work in teams of 3 - 4 students to complete the assignment.

Endpoint detection and response (EDR) is a cybersecurity technology that monitors endpoints, such as servers, laptops, and Internet-of-Things devices, to mitigate malicious cyber threats.

SIEM stands for Security Information and Event Management. It's a field of computer security that combines Security Information Management (SIM) and Security Event Management (SEM). SIEM tools provide real-time analysis of security alerts generated by endpoints, applications and network hardware.

Wazuh is an open-source security platform that provides unified EDR and SIEM protection for endpoints and cloud workloads. SIEM and EDR platforms are commonly used in cybersecurity to enhance the security posture of organizations.

Tasks

Requirements

Setup the following requirements to complete the tasks in this lab:

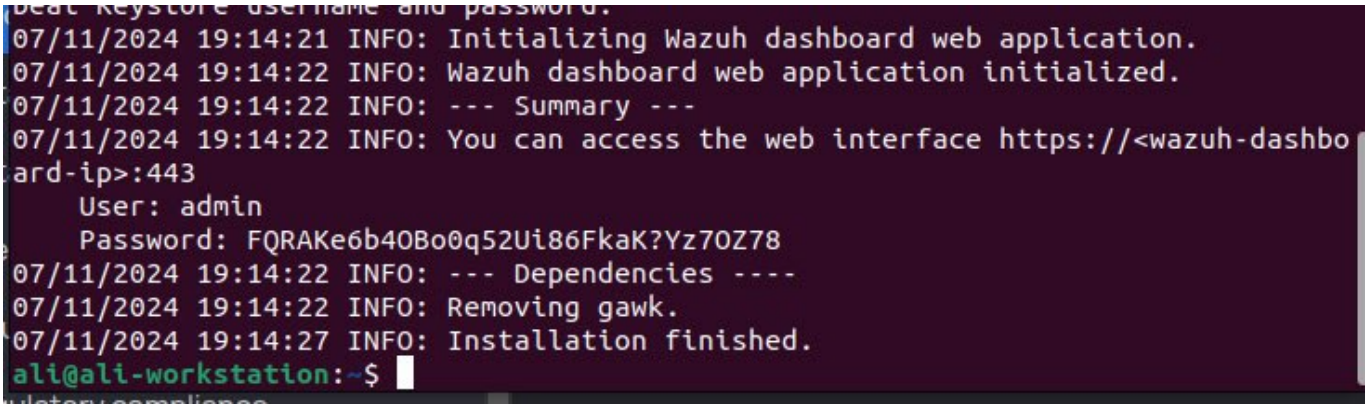
- Deploy the Wazuh central components on a Linux endpoint using the Quickstart installation guide.

You can also deploy Wazuh with OVA virtual machine.

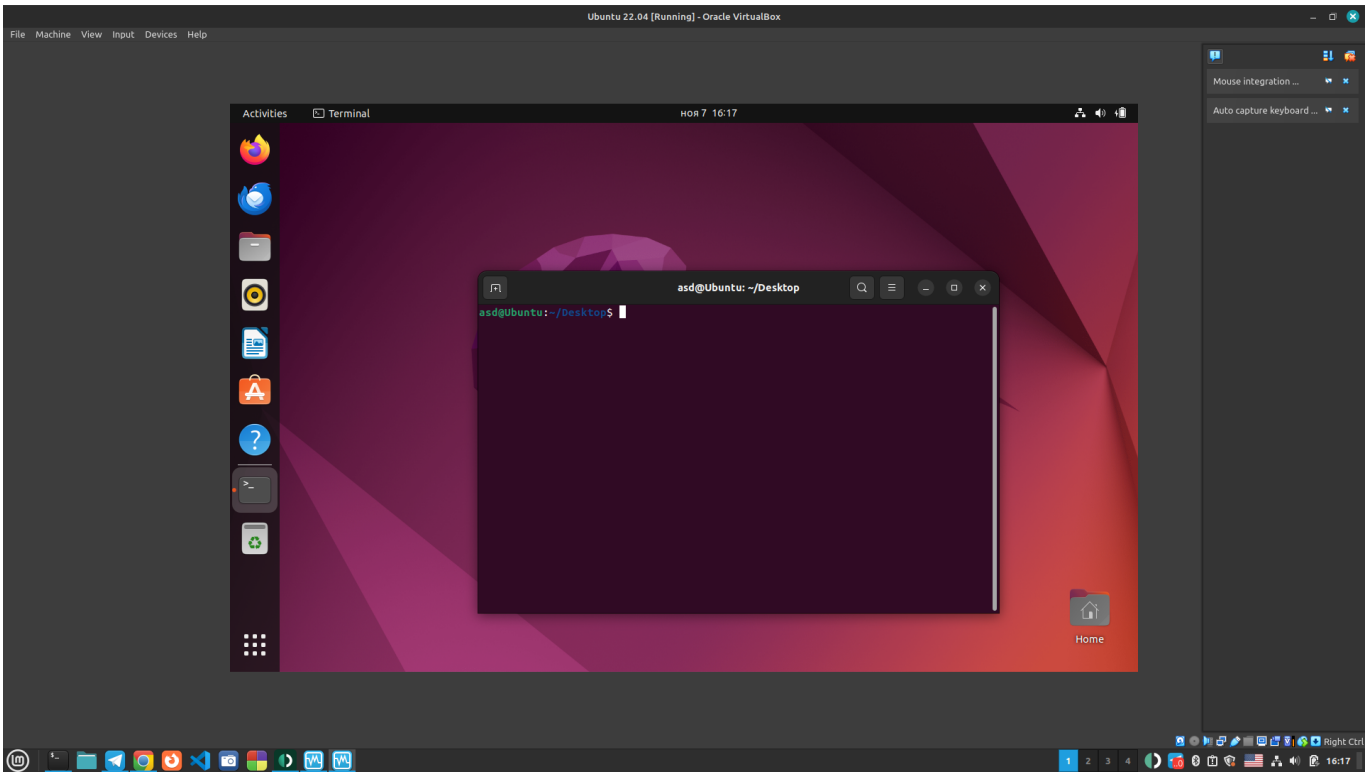
- Deploy the Wazuh agent on a second Linux endpoint. This will be the monitored endpoint you aim to protect.

Solution:

- I installed the server on ubuntu device:



- I used Ubuntu 22.04 as client agent, here is after install it in Virtualbox



- I added the Wazuh repo and install it.

```
root@Ubuntu:/home/asd# curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | gpg --no-default-keyring --keyring gnupg-ring:/usr/sha
re/keyrings/wazuh.gpg --import && chmod 644 /usr/share/keyrings/wazuh.gpg
gpg: keyring '/usr/share/keyrings/wazuh.gpg' created
gpg: directory '/root/.gnupg' created
gpg: /root/.gnupg/trustdb.gpg: trustdb created
gpg: key 96B3EE5F29111145: public key "Wazuh.com (Wazuh Signing Key) <support@wazuh.com>" imported
gpg: Total number processed: 1
gpg:               imported: 1
root@Ubuntu:/home/asd# echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable main" | tee -a
 /etc/apt/sources.list.d/wazuh.list
deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable main
root@Ubuntu:/home/asd# apt-get update
Hit:1 http://gb.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://gb.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Hit:4 http://gb.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:5 https://packages.wazuh.com/4.x/apt stable InRelease [17,3 kB]
Get:6 https://packages.wazuh.com/4.x/apt stable/main amd64 Packages [41,6 kB]
Get:7 https://packages.wazuh.com/4.x/apt stable/main i386 Packages [12,0 kB]
Fetched 328 kB in 11s (28,7 kB/s)
Reading package lists... Done
root@Ubuntu:/home/asd#
```

- Deploy wazuh agent with specifying the ip address of the server:

```
10.100.20.55
```

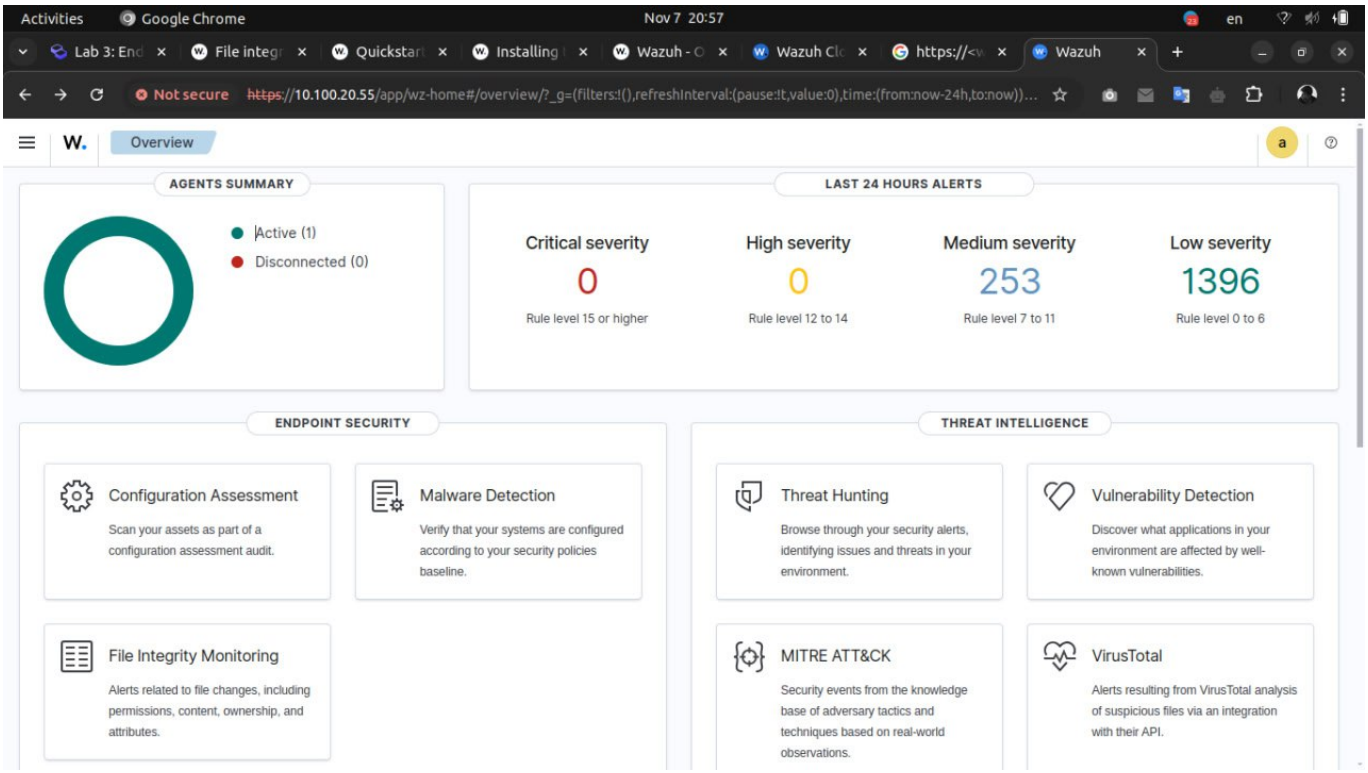
```
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500
(UNSPEC)
        RX packets 304338  bytes 416494573 (416.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 130663  bytes 22398566 (22.3 MB)
        TX errors 0  dropped 1729 overruns 0  carrier 0  collisions 0

virbr0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 192.168.122.1  netmask 255.255.255.0  broadcast 192.168.122.255
        ether 52:54:00:e2:29:b4  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlp7s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.100.20.55  netmask 255.255.254.0  broadcast 10.100.21.255
        inet6 fe80::2894:893d:805e:b6f  prefixlen 64  scopeid 0x20<link>
        ether d0:53:49:c0:25:94  txqueuelen 1000  (Ethernet)
        RX packets 255  bytes 44169 (44.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 1630
        TX packets 420  bytes 54841 (54.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 18

ali@ali-workstation:~$
```

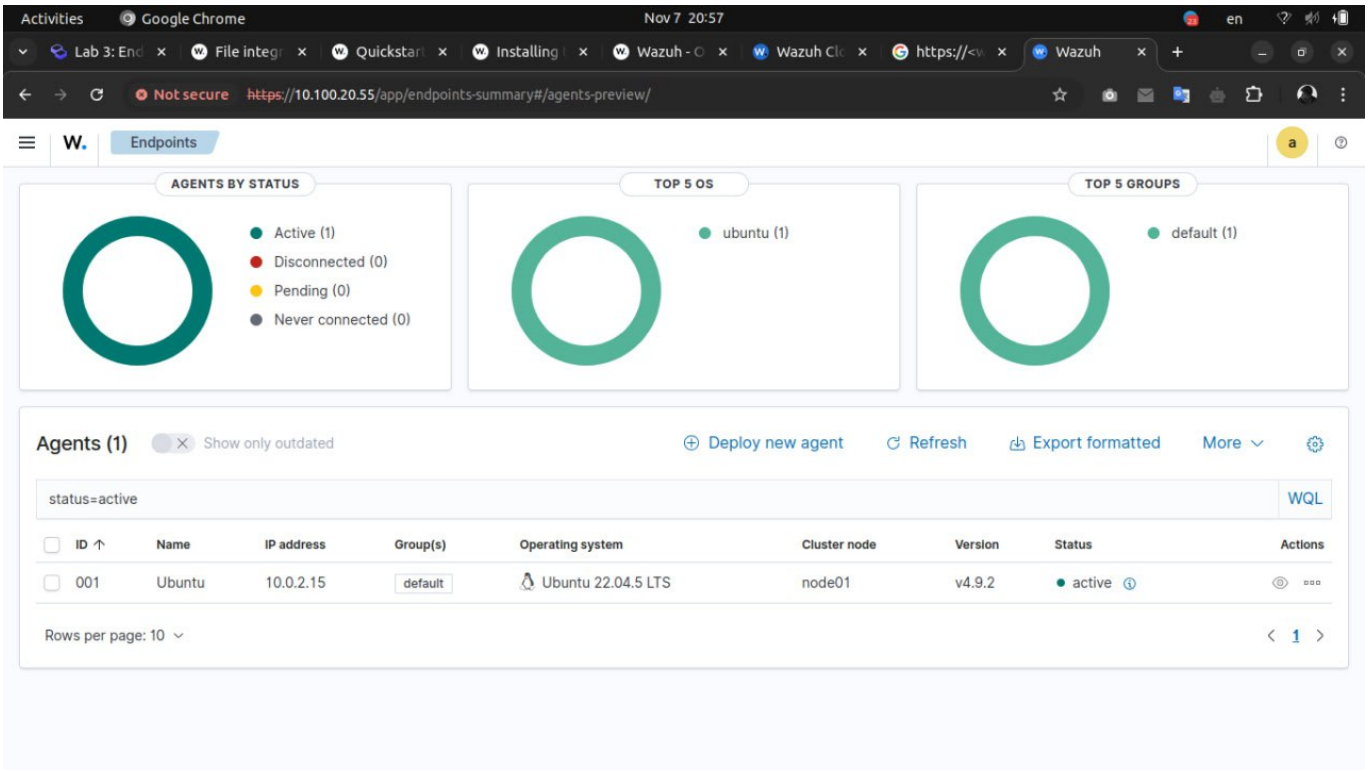
```
asd@Ubuntu:~/Desktop$ sudo WAZUH_MANAGER="10.100.20.55" apt-get install wazuh-agent
[sudo] password for asd:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  wazuh-agent
0 upgraded, 1 newly installed, 0 to remove and 99 not upgraded.
Need to get 10,8 MB of archives.
After this operation, 37,3 MB of additional disk space will be used.
Get:1 https://packages.wazuh.com/4.x/apt stable/main amd64 wazuh-agent amd64 4.9.2-1 [10,8 MB]
Fetched 10,8 MB in 2s (6 059 kB/s)
Preconfiguring packages ...
Selecting previously unselected package wazuh-agent.
(Reading database ... 201764 files and directories currently installed.)
Preparing to unpack .../wazuh-agent_4.9.2-1_amd64.deb ...
Unpacking wazuh-agent (4.9.2-1) ...
Setting up wazuh-agent (4.9.2-1) ...
```

- Enable and start Wazuh agent service

```
asd@Ubuntu:~/Desktop$ systemctl daemon-reload
asd@Ubuntu:~/Desktop$ systemctl enable wazuh-agent
Created symlink /etc/systemd/system/multi-user.target.wants/wazuh-agent.service → /lib/systemd/system/wazuh-agent.service.
asd@Ubuntu:~/Desktop$ systemctl start wazuh-agent
asd@Ubuntu:~/Desktop$
```

- finally, you can see the dashboard with my agent inforamtion:





1. File integrity monitoring

- Explain the importance of file integrity monitoring.
- Configure the agent on your monitored endpoint to perform file integrity monitoring. Monitor any directory or file of your choice and show alerts on the Wazuh dashboard when changes are made to the monitored directory or file.

Follow the FIM PoC described in the Wazuh official documentation.

Solution:

- File integrity monitoring (FIM) is crucial for spotting weak points in systems that hackers could exploit. It helps organizations address issues early, strengthens defenses, and aids in regulatory compliance, reducing risks of fines and operational disruptions. FIM also guides security teams to focus on critical risks, making overall protection more effective.

I added this configruation to the file osscf.config

```
<directories check_all="yes" report_changes="yes"
realtime="yes">/home</directories>
```

and updated the interval to run the testing each 5m

```
<interval>5m</interval>
```

then restart the service:

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

```
<wodle name="syscollector">
  <disabled>no</disabled>
  <interval>5m</interval>
  <scan_on_start>yes</scan_on_start>
  <hardware>yes</hardware>
  <os>yes</os>
  <network>yes</network>
  <packages>yes</packages>
  <ports all="no">yes</ports>
  <processes>yes</processes>

  <!-- Database synchronization settings -->
  <synchronization>
    <max_eps>10</max_eps>
  </synchronization>
</wodle>

<sca>
  <enabled>yes</enabled>
  <scan_on_start>yes</scan_on_start>
  <interval>12h</interval>
  <skip_nfs>yes</skip_nfs>
</sca>

<!-- File integrity monitoring -->
<syscheck>
  <disabled>no</disabled>
  <directories check_all="yes" report_changes="yes" realtime="yes">/home</directories>
```

create new file:

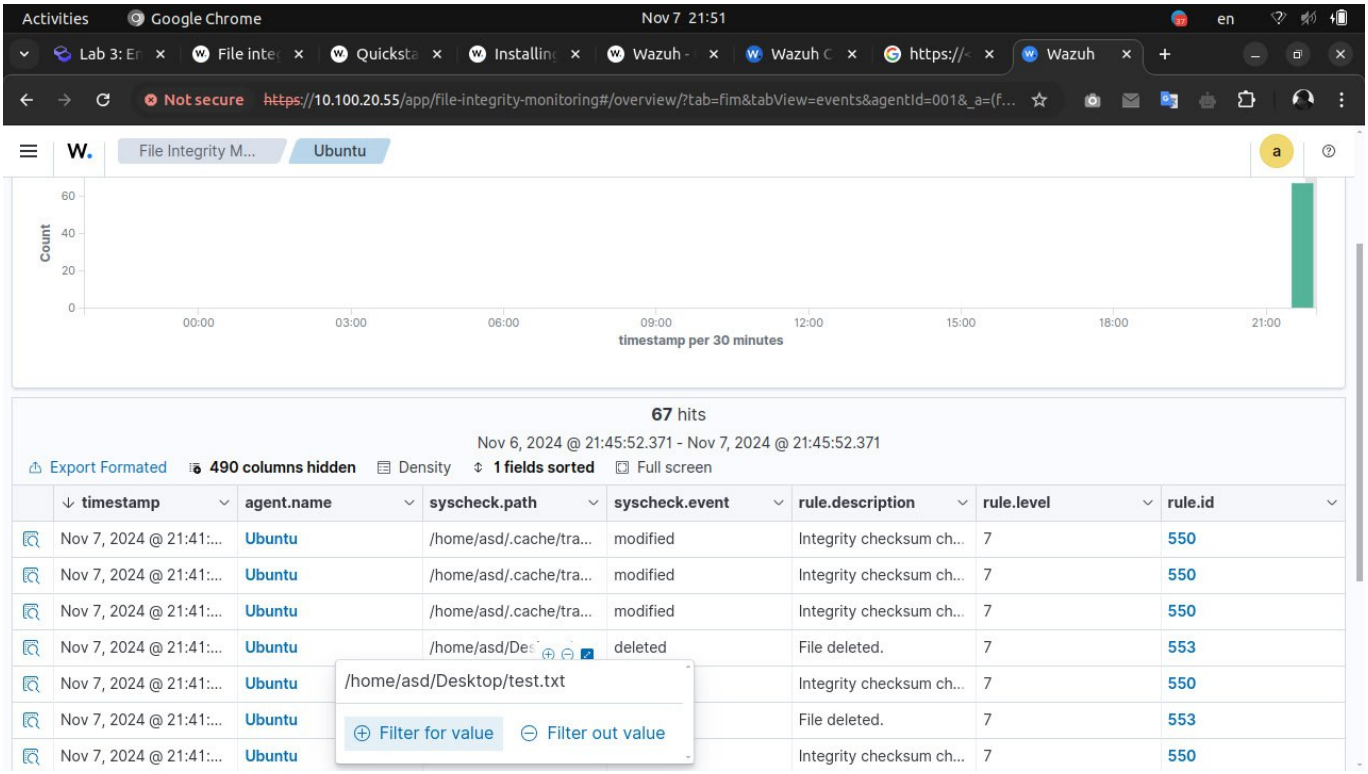
```
asd@Ubuntu:~/Desktop$ nano /home/asd/Desktop/test1.txt
asd@Ubuntu:~/Desktop$
```

delete file:

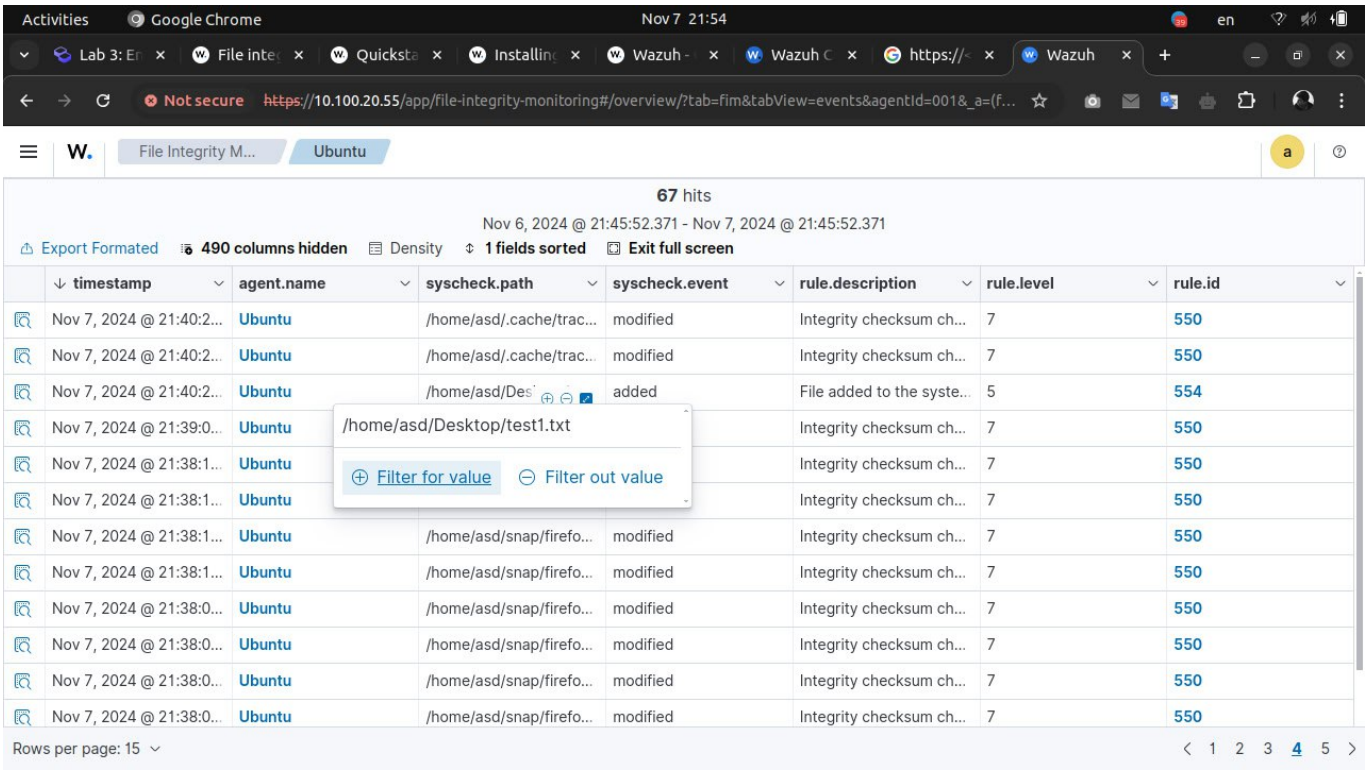
```
asd@Ubuntu:~/Desktop$ rm /home/asd/Desktop/test.txt
asd@Ubuntu:~/Desktop$
```

from the dashboard:

Deleted file:



added file:



2. Vulnerability detection

- Describe the significance of vulnerability detection.
- What are the differences between authenticated and unauthenticated vulnerability scans?
- Why do we need to perform periodic vulnerability scans?
- On the Wazuh dashboard, show the vulnerabilities discovered on the monitored endpoint.

  Install at least two vulnerable apps if none is discovered on your default installation.
- Patch at least two vulnerabilities on the endpoint and show updates on the Wazuh dashboard to prove that the patch worked.

  You may need to wait an hour for the next system inventory update to show results of the patch.
- Which type of scan is Wazuh vulnerability detection scan more closely related with (authenticated or unauthenticated)?

Give reasons for your answer.

Solution:

1. Vulnerability detection finds weaknesses that attackers could exploit. By addressing these early, companies can prevent issues, strengthen security, stay compliant, and avoid unexpected downtime. It also allows security teams to prioritize the most critical threats, ensuring more focused protection.

---

2. **Authenticated Scans**: Use login credentials to access deeper internal details, improving accuracy and reducing false positives. They reveal vulnerabilities that attackers could exploit from an internal perspective.

   Unauthenticated Scans: Run without credentials, showing only what an external attacker would see (e.g., open ports). They are useful for finding publicly exposed weaknesses.

   In short, authenticated scans provide an internal view, while unauthenticated scans offer an external attacker's perspective. Both are essential for a comprehensive security assessment.

---

3. **Spot New Threats**: Periodic scans catch emerging vulnerabilities.

   Adapt to System Changes: Scans reveal issues from updates or configuration changes.

   Meet Compliance Standards: Regular scans ensure compliance with security regulations.

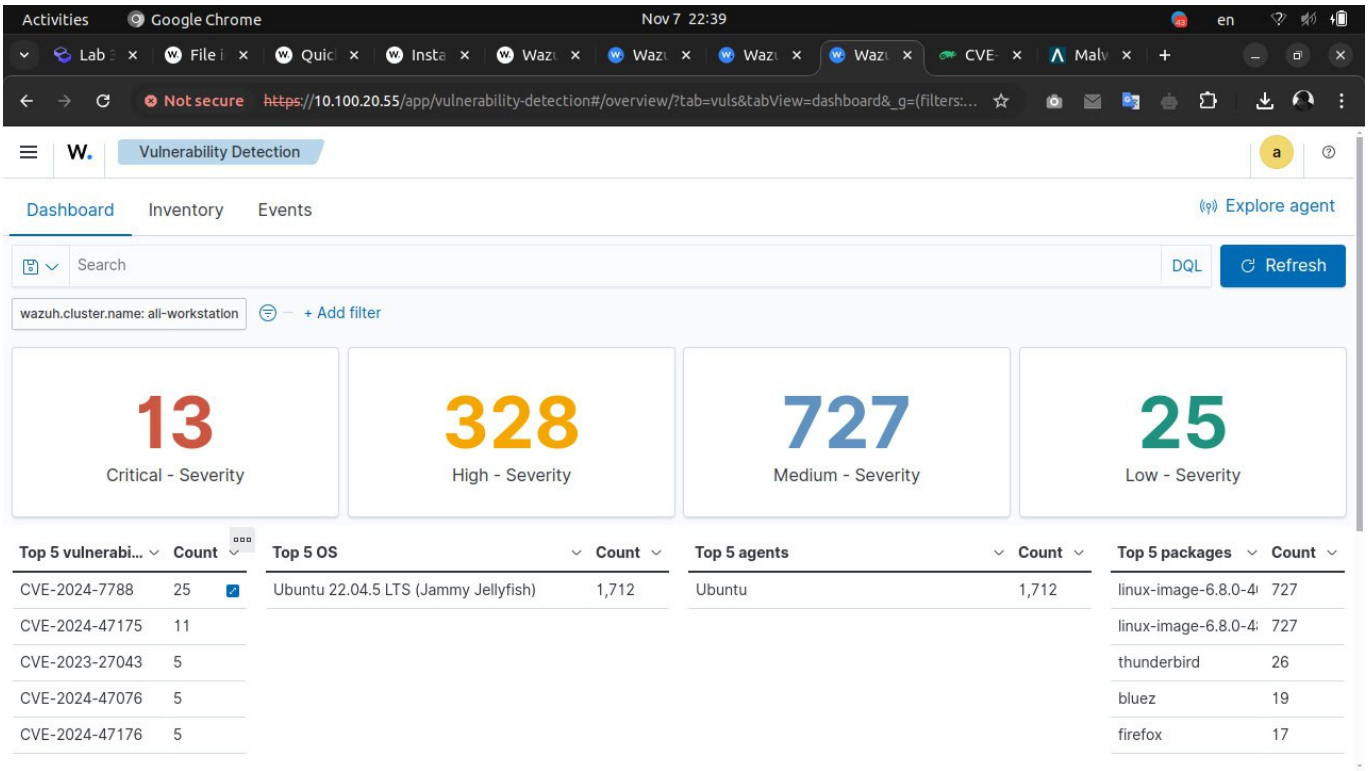
   Prevent Breaches: Scans detect vulnerabilities before they can be exploited.

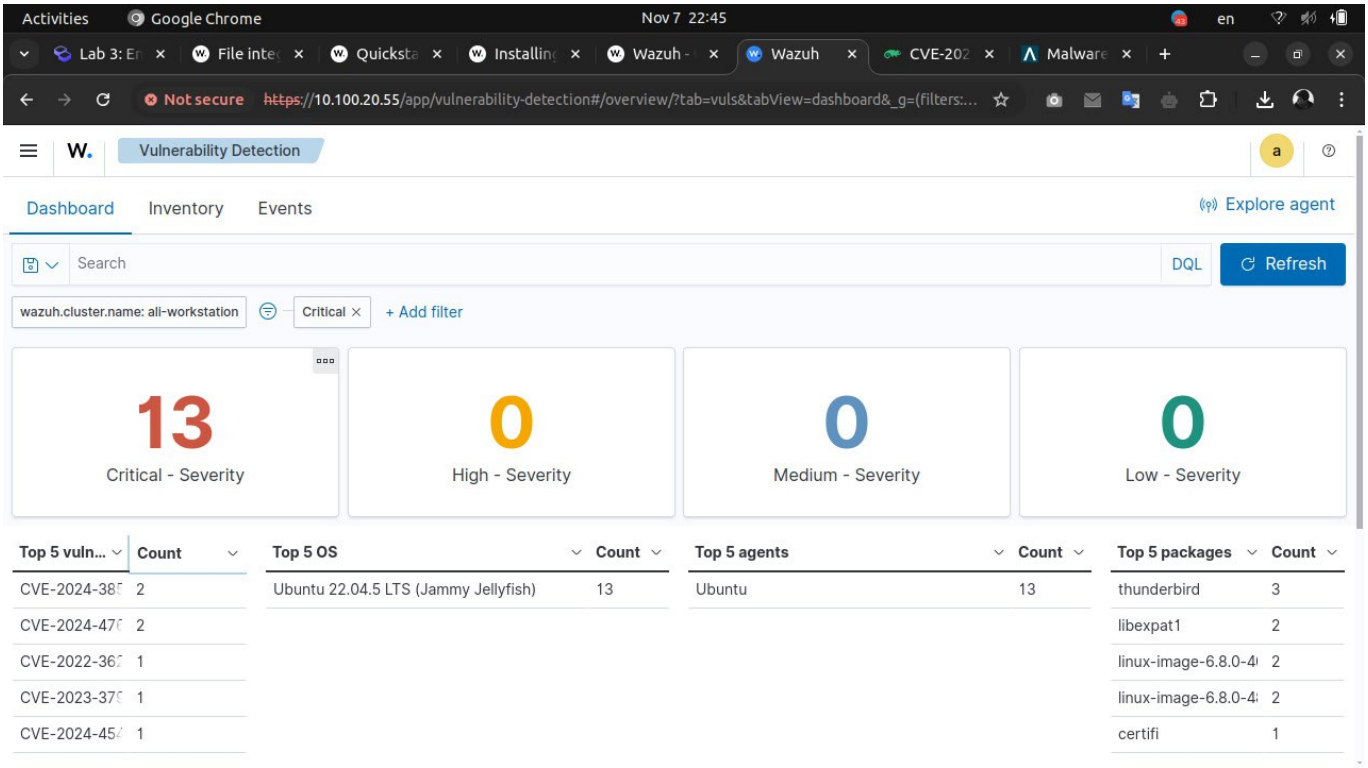
   Stay Proactive: Regular scanning helps close security gaps and strengthen defenses.

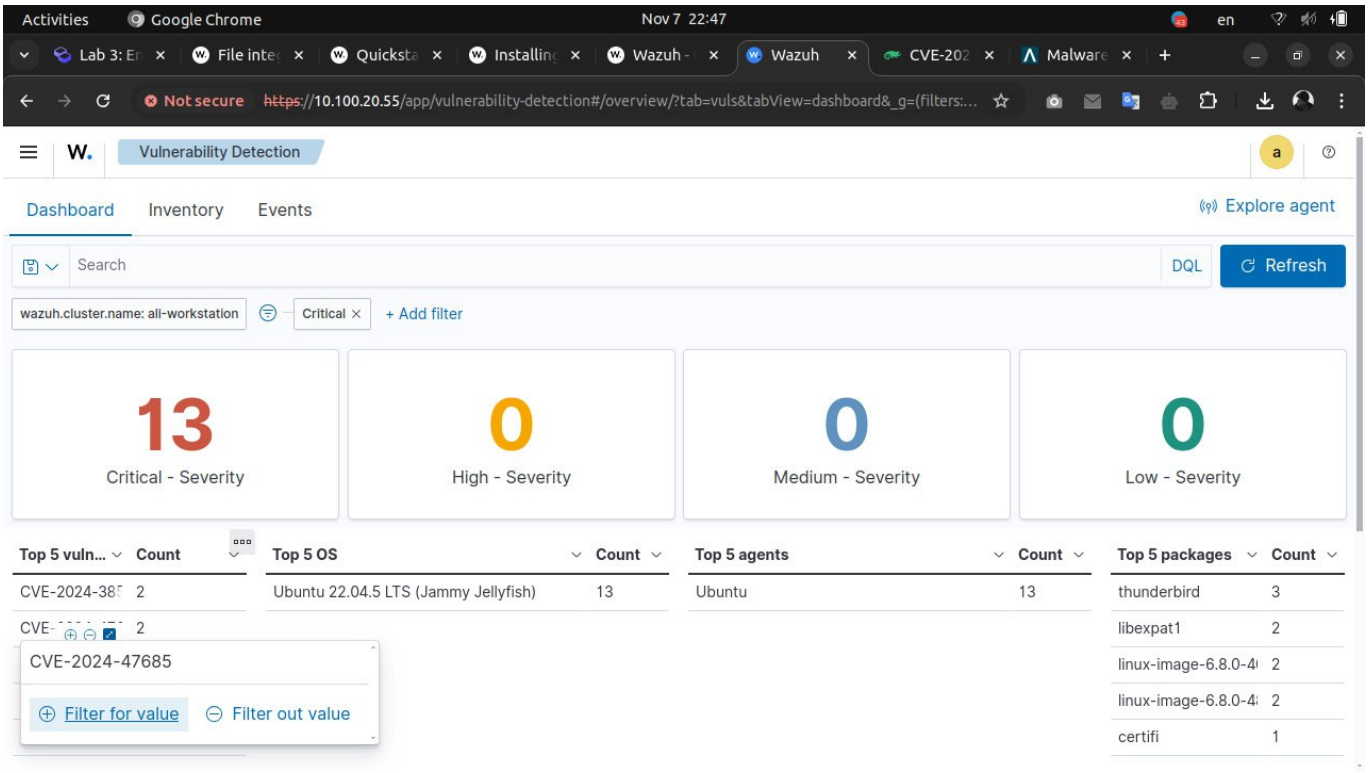
   Regular scans are key to staying secure, compliant, and ahead of threats.

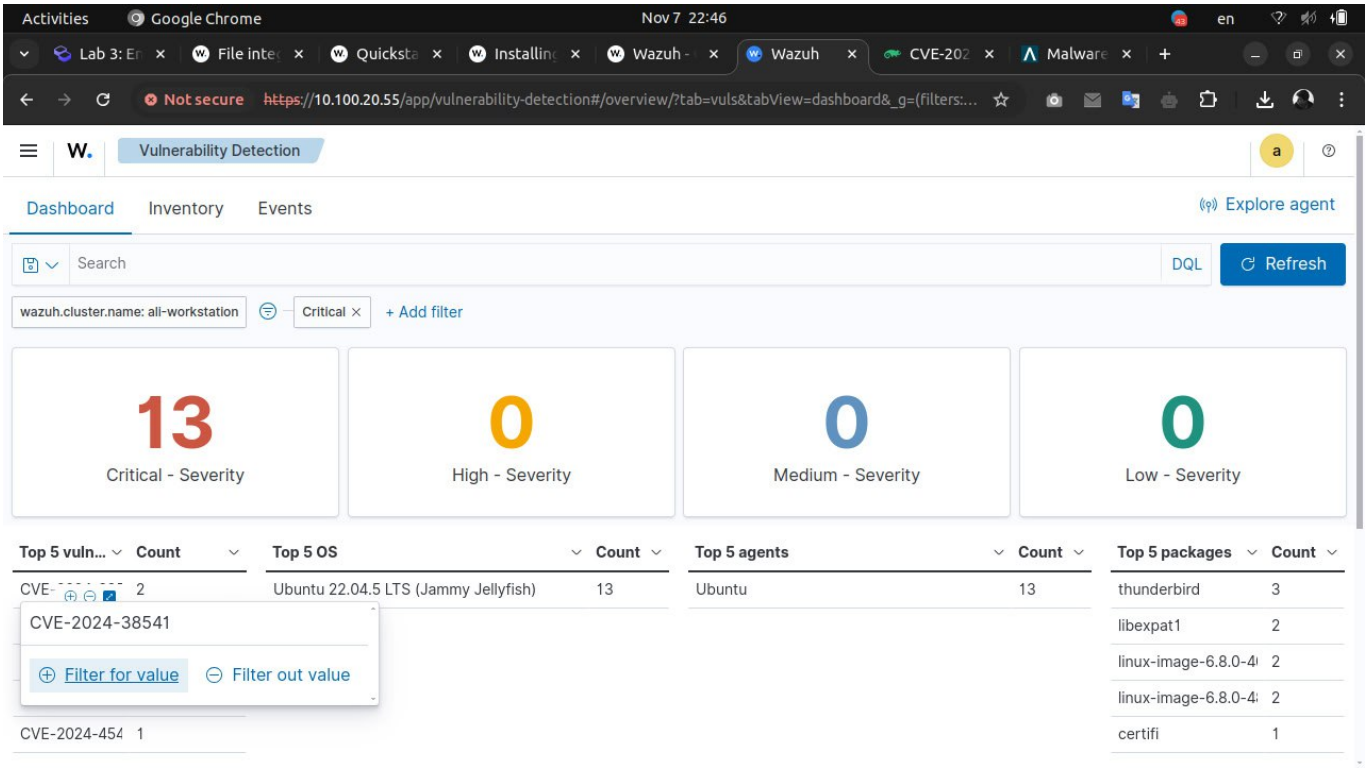
---

4. When checking the Dashboard I fonund:


Activities
Google Chrome
Nov 7 22:39
en
Not secure
https://10.100.20.55/app/vulnerability-detection#/overview/?tab=vuls&tabView=dashboard&_g=(filters:...
Vulnerability Detection
Dashboard
Inventory
Events
Explore agent
Search
DQL
Refresh
wazuh.cluster.name: all-workstation
+ Add filter
13
Critical - Severity
328
High - Severity
727
Medium - Severity
25
Low - Severity
Top 5 vulnerabi...
Count
CVE-2024-7788 25
CVE-2024-47175 11
CVE-2023-27043 5
CVE-2024-47076 5
CVE-2024-47176 5
Top 5 OS
Count
Ubuntu 22.04.5 LTS (Jammy Jellyfish) 1,712
Top 5 agents
Count
Ubuntu 1,712
Top 5 packages
Count
linux-image-6.8.0-4 727
linux-image-6.8.0-4 727
thunderbird 26
bluez 19
firefox 17

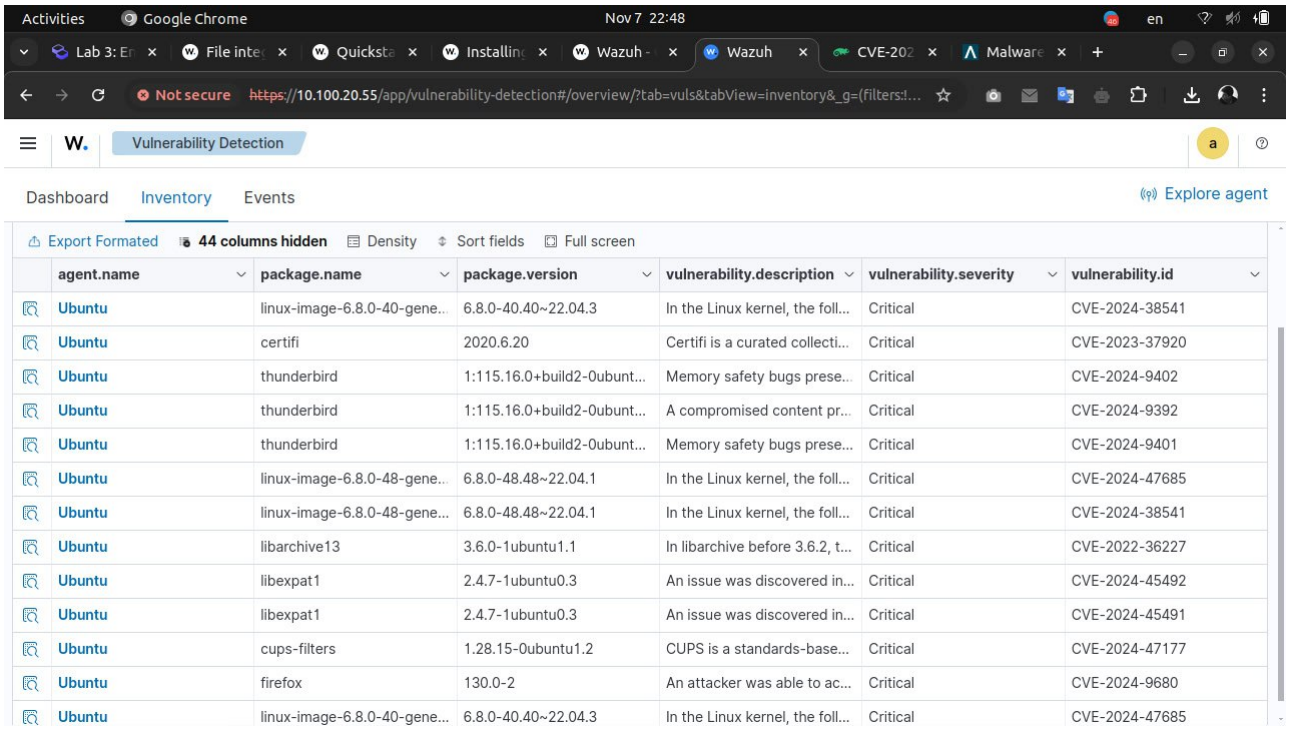

Activities
Google Chrome
Nov 7 22:45
en
Not secure
https://10.100.20.55/app/vulnerability-detection#/overview/?tab=vuls&tabView=dashboard&_g=(filters:...
Vulnerability Detection
Dashboard
Inventory
Events
Explore agent
Search
DQL
Refresh
wazuh.cluster.name: all-workstation
Critical
+ Add filter
13
Critical - Severity
0
High - Severity
0
Medium - Severity
0
Low - Severity
Top 5 vuln...
Count
CVE-2024-38 2
CVE-2024-47 2
CVE-2022-36 1
CVE-2023-37 1
CVE-2024-45 1
Top 5 OS
Count
Ubuntu 22.04.5 LTS (Jammy Jellyfish) 13
Top 5 agents
Count
Ubuntu 13
Top 5 packages
Count
thunderbird 3
libexpat1 2
linux-image-6.8.0-4 2
linux-image-6.8.0-4 2
certifi 1

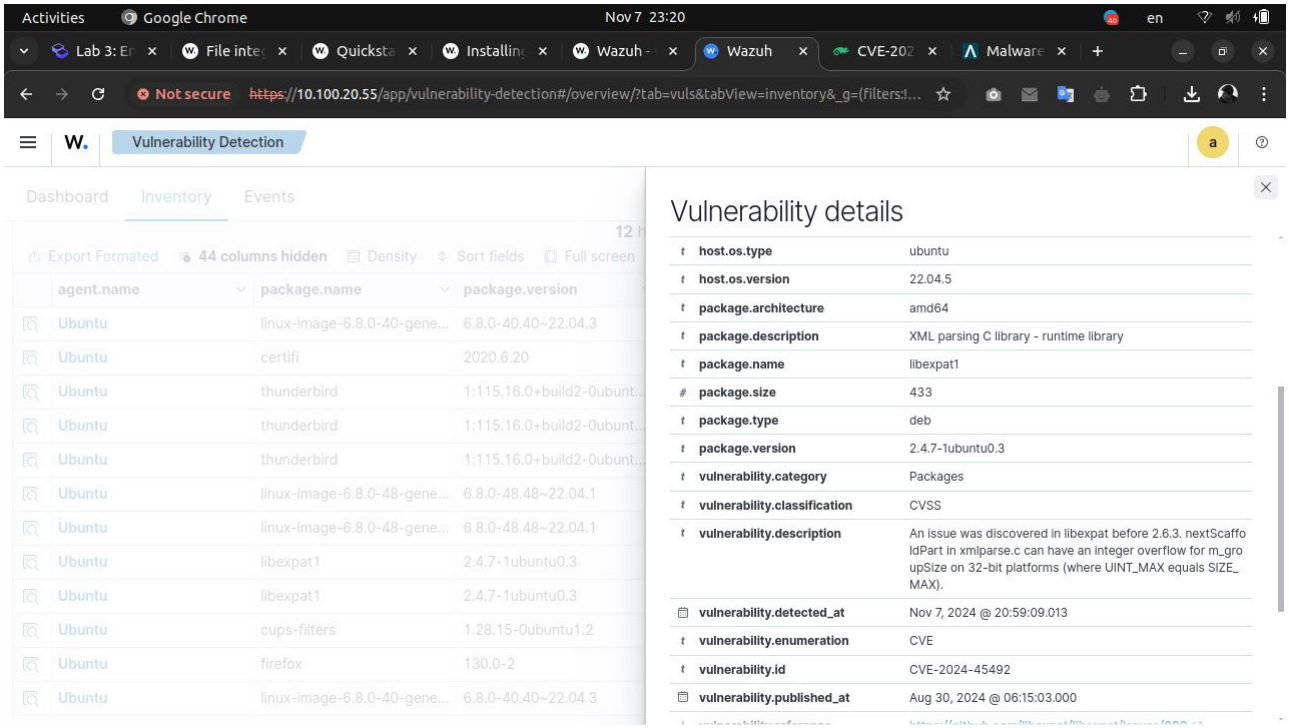


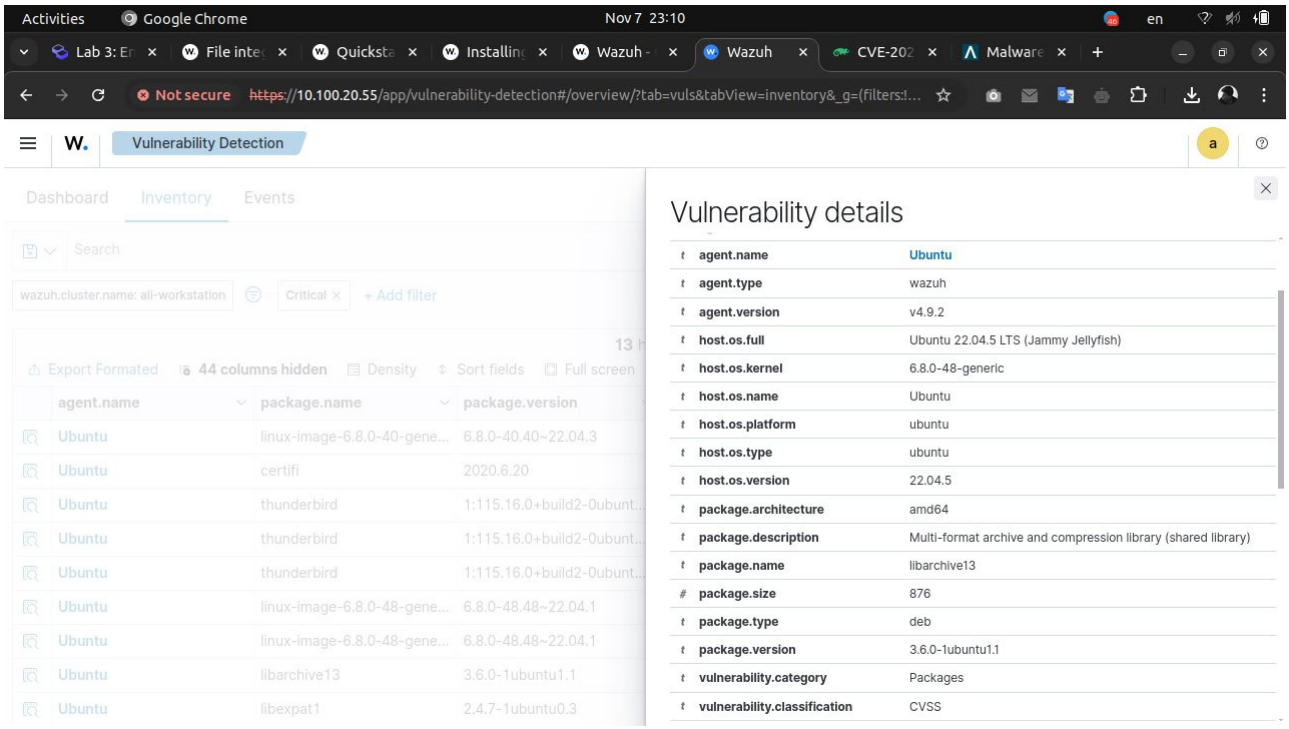


5. I updated two of the packges `libarchieve3` and `libexpat1`

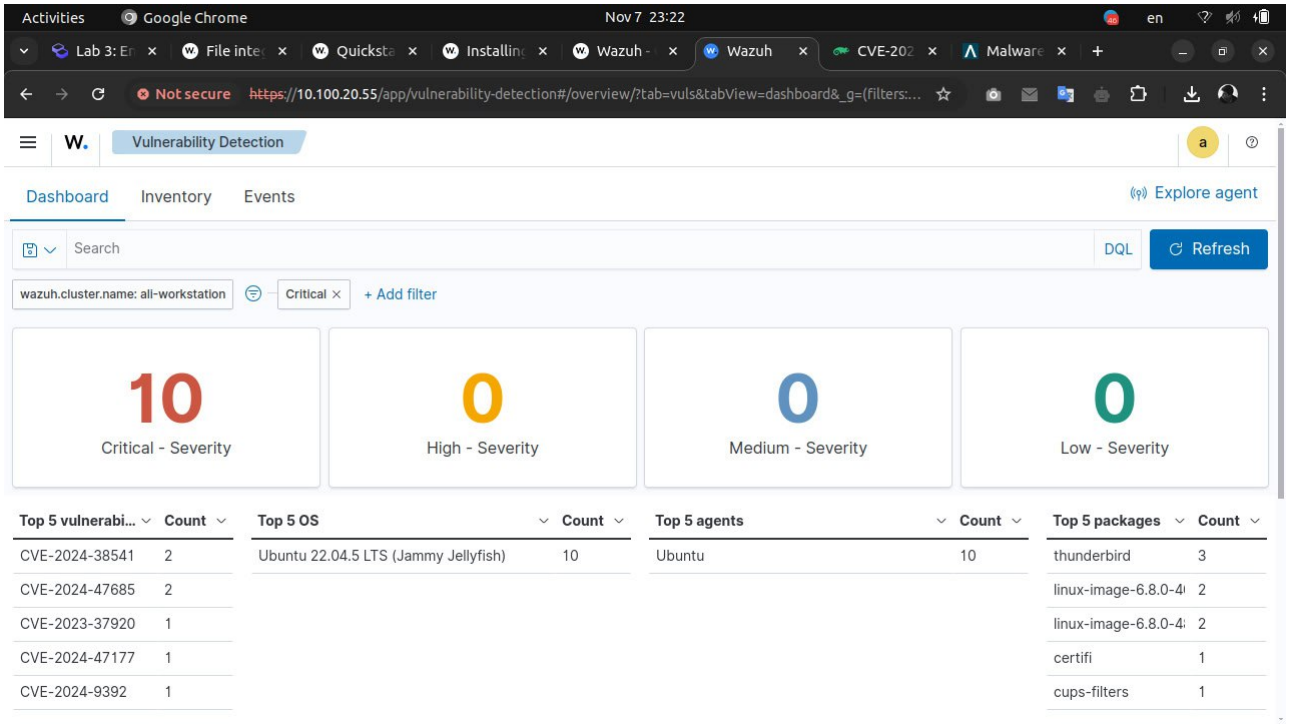
```
asd@Ubuntu:~/Desktop$ sudo apt install --upgrade libarchive13
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  lrzip
The following packages will be upgraded:
  libarchive13
1 upgraded, 0 newly installed, 0 to remove and 98 not upgraded.
Need to get 0 B/369 kB of archives.
After this operation, 0 B of additional disk space will be used.
(Reading database ... 210712 files and directories currently installed.)
Preparing to unpack .../libarchive13_3.6.0-1ubuntu1.3_amd64.deb ...
Unpacking libarchive13:amd64 (3.6.0-1ubuntu1.3) over (3.6.0-1ubuntu1.1) ...
Setting up libarchive13:amd64 (3.6.0-1ubuntu1.3) ...
Processing triggers for libc-bin (2.35-0ubuntu3.8) ...
asd@Ubuntu:~/Desktop$
```







after updating we notice the removing of critical issues



---

6. Wazuh vulnerability detection scans are more like **authenticated scans** because:

    1. **Direct System Access**: Wazuh agents are installed on monitored systems, accessing internal details.

    2. **Detailed Scanning**: This internal access enables deeper scanning.

    3. **Better Accuracy**: With direct access, Wazuh reduces false positives.

---

3. Detecting cryptominers

Refer to Detecting illegitimate crypto miners on Linux endpoints for guidance.

- What are cryptominers?
- Are cryptominers inherently malicious? Justify your answer.
- What are the generic steps through which Linux cryptomining botnets operate?
- Configure your Wazuh server to detect at least two of the steps highlighted above.
- Trigger alerts for the rules you have created by simulating the cryptominer behavior you are detecting.
- Provide explanation for any regular expression used in your rule.

Official documentation of Wazuh rule syntax.

You can also analyze the full event log by expanding the alert on the Wazuh dashboard. This will help understand the rule better.

Solution:

1. Cryptominers are programs that use computing power to mine cryptocurrency by solving blockchain problems. While some mining is legal, **cryptojacking** is the illegal practice of installing mining software without permission, leading to slow systems and increased energy costs.

   Detecting cryptominers on Linux involves monitoring for unusual activity, like high CPU usage or unknown processes, which may indicate unauthorized mining.

---

2. Cryptominers aren't inherently malicious; they're tools for mining cryptocurrency. However, they become malicious if installed without permission, a practice known as cryptojacking, which leads to slow performance, high energy costs, and possible hardware damage.

---

3. Linux cryptomining botnets usually operate through these steps:
   - **Infection**: Attackers gain access through system vulnerabilities or phishing.
   - **Deploying the Miner**: Cryptomining software is installed.
   - **Establishing Persistence**: Cron jobs or startup scripts ensure the miner runs after reboots.
   - **Using System Resources**: The miner uses CPU and GPU power, often hidden in the background.
   - **Spreading**: The botnet tries to infect other devices on the network.
   - **Connecting for Updates**: The botnet communicates with a server for updates or to adjust mining settings.

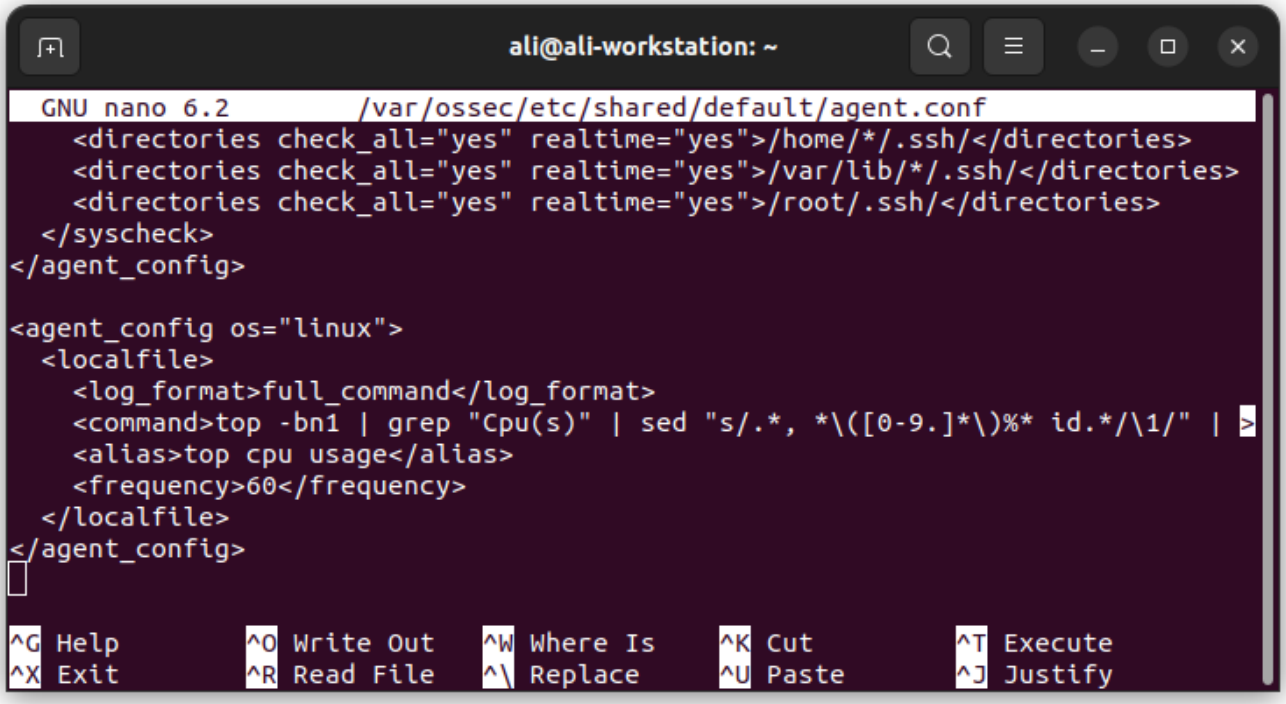
---

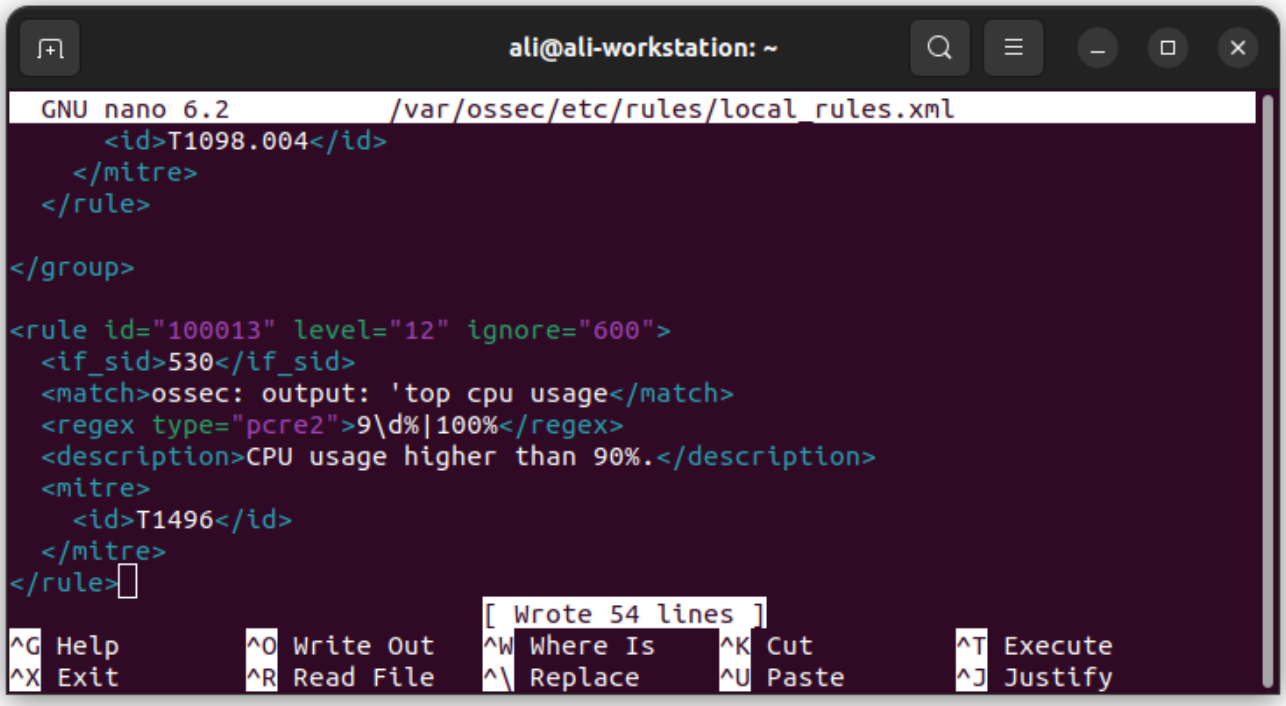
4. open the config file:

we configure the changing of authorized_keys, cpu usage, and ssh connection failures.

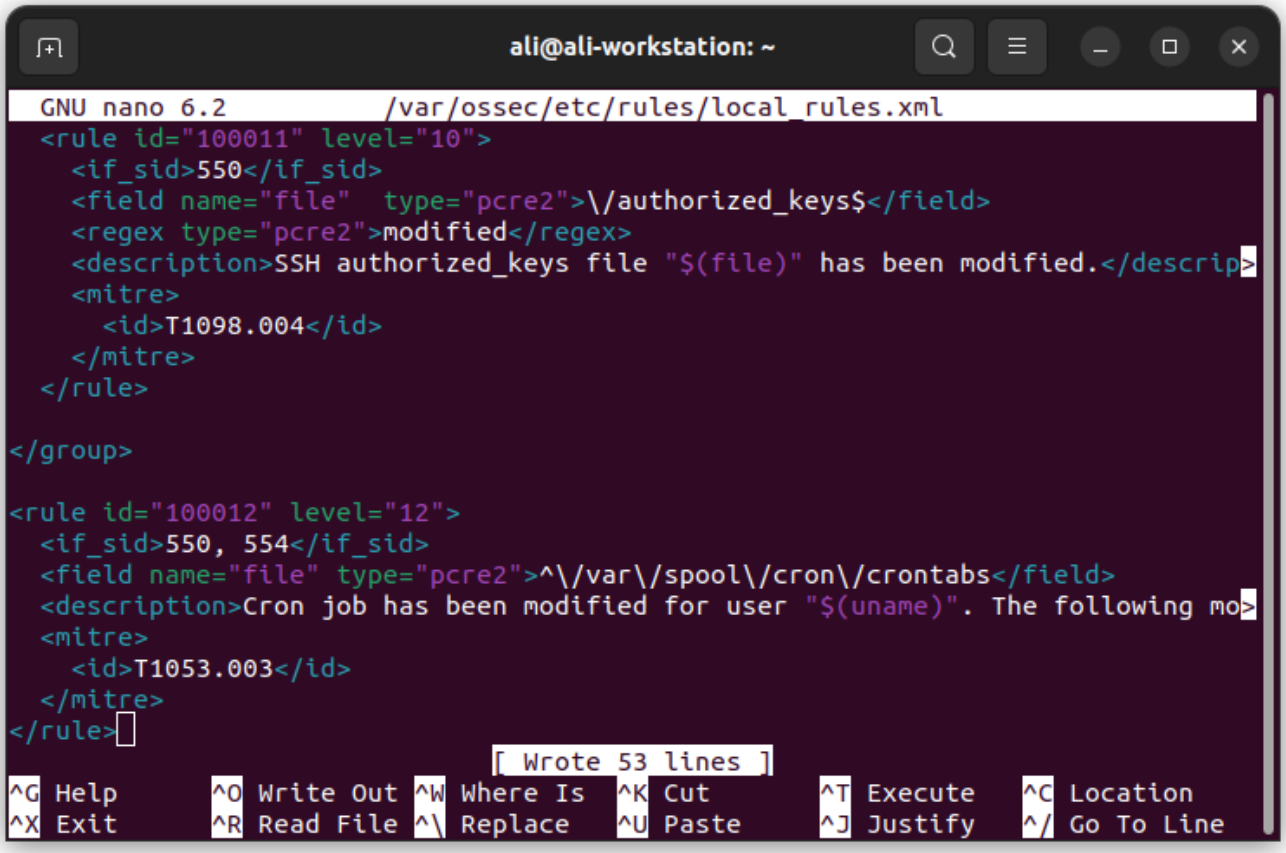
```
ali@ali-workstation:~$ sudo nano /var/ossec/etc/shared/default/agent.conf
ali@ali-workstation:~$ sudo nano /var/ossec/etc/shared/default/agent.conf
ali@ali-workstation:~$
```

CPU Usage:





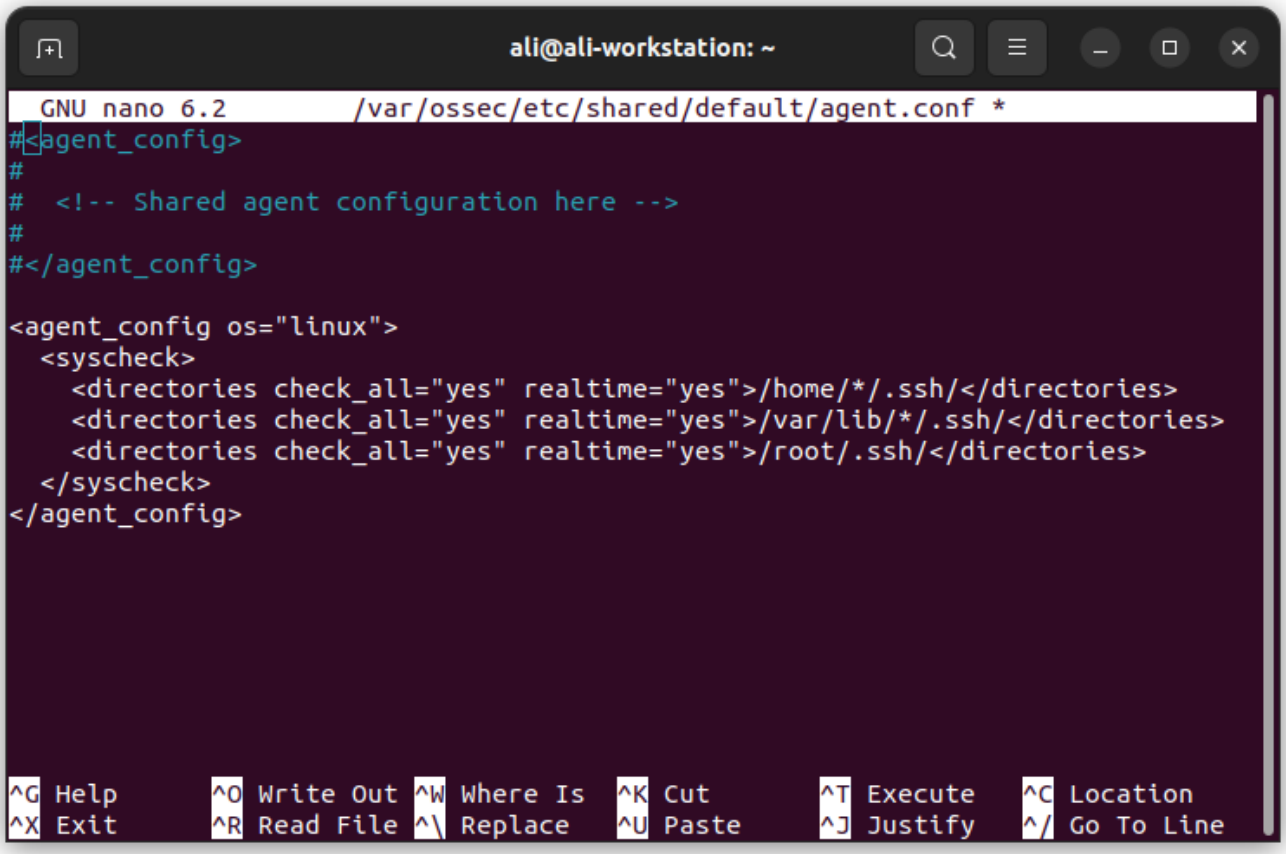
authorized_keys file:



on the agent:

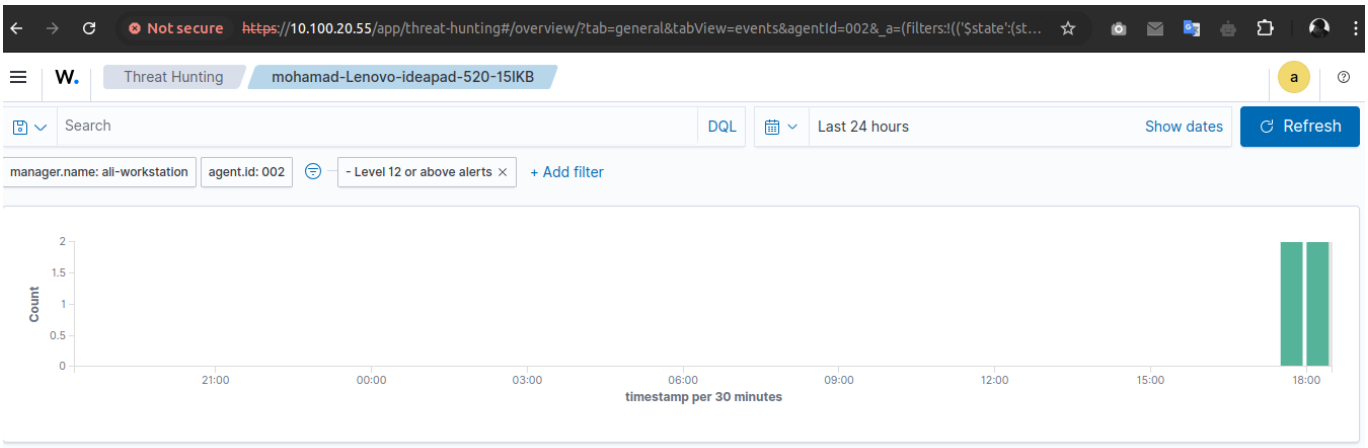
```
ohamad@mohamad-Lenovo-ideapad-520-15IKB:~$ cat /var/ossec/etc/local_internal_op
tions.conf
cat: /var/ossec/etc/local_internal_options.conf: Permission denied
ohamad@mohamad-Lenovo-ideapad-520-15IKB:~$ sudo cat /var/ossec/etc/local_intern
l_options.conf
[sudo] password for mohamad:
# local_internal_options.conf
#
# This file should be handled with care. It contains
# run time modifications that can affect the use
# of OSSEC. Only change it if you know what you
# are doing. Look first at ossec.conf
# for most of the things you want to change.
#
# This file will not be overwritten during upgrades.
logcollector.remote_commands=1
ohamad@mohamad-Lenovo-ideapad-520-15IKB:~$
```

ssh connections:



5.

results on dashboard after alerts:

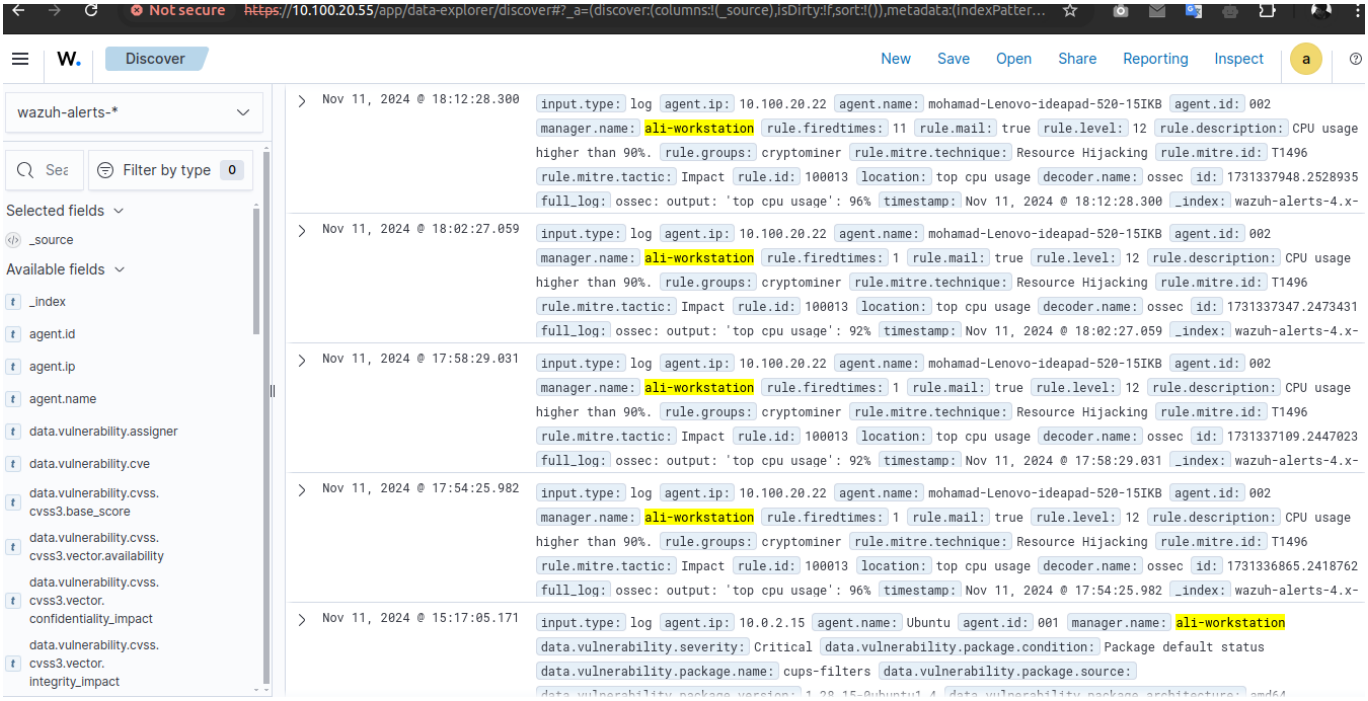


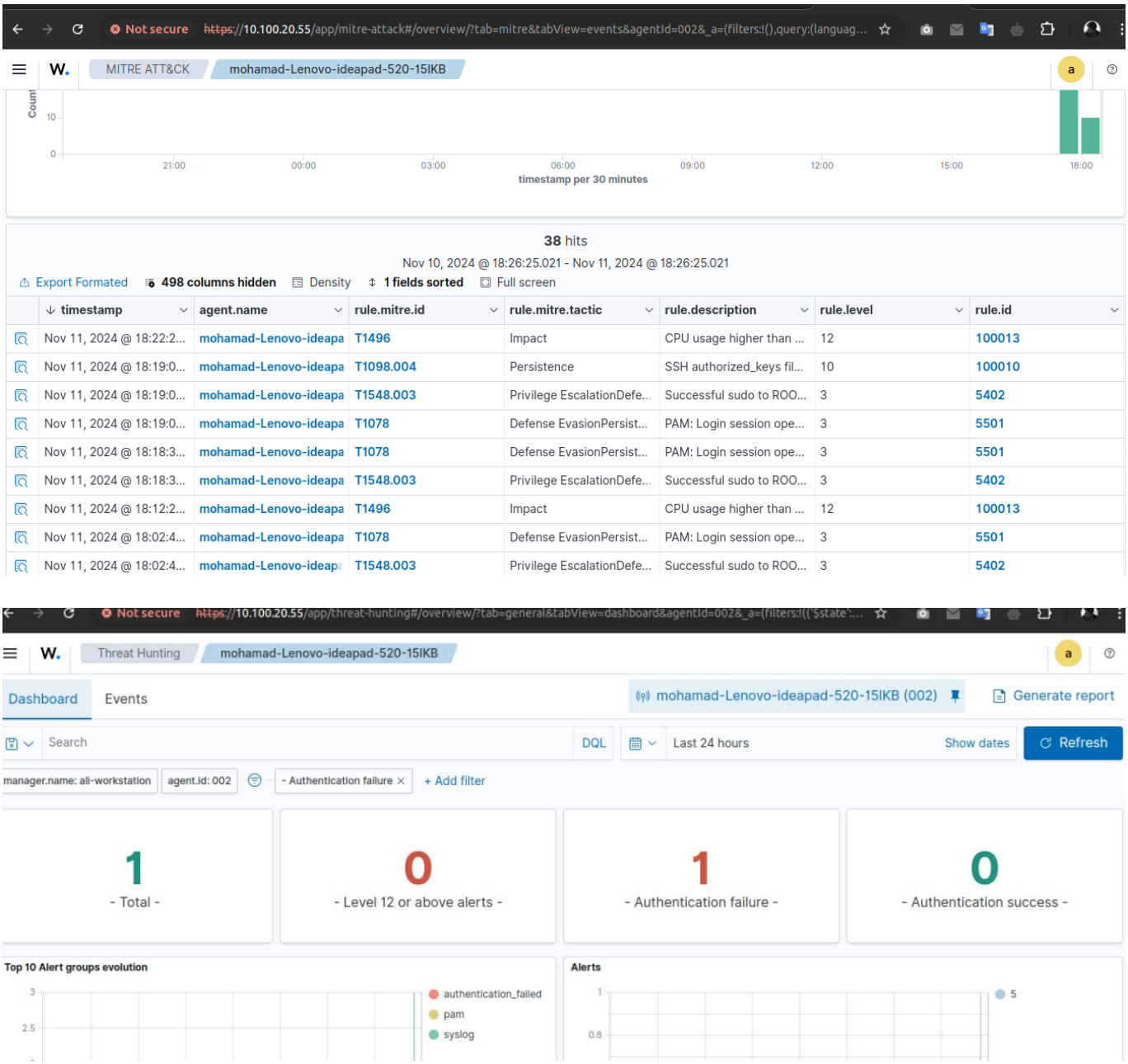
4 hits

Nov 10, 2024 @ 18:17:39.372 - Nov 11, 2024 @ 18:17:39.372

Export Formated | **500 columns hidden** | Density | **1 fields sorted** | Full screen

timestamp	agent.name	rule.description	rule.level	rule.id
Nov 11, 2024 @ 18:12:28.300	mohamad-Lenovo-ideapad-520-15II	CPU usage higher than 90%.	12	100013
Nov 11, 2024 @ 18:02:27.059	mohamad-Lenovo-ideapad-520-15II	CPU usage higher than 90%.	12	100013
Nov 11, 2024 @ 17:58:29.031	mohamad-Lenovo-ideapad-520-15II	CPU usage higher than 90%.	12	100013
Nov 11, 2024 @ 17:54:25.982	mohamad-Lenovo-ideapad-520-15II	CPU usage higher than 90%.	12	100013





---

6. `\/authorized_keys$`
    - This pattern matches any file path ending in `authorized_keys`.
    - *Explanation*:
        - `\/`: Matches the `/` character in the file path.
        - `authorized_keys`: Matches the filename "authorized_keys".
        - `$`: Asserts that this is the end of the file path.

This regex helps detect changes to the `authorized_keys` file, used for SSH access control.
