

# iu-cf-lab-04-Iskander\_Nafikov

- **Name:** Iskander Nafikov
  - **E-mail:** [i.nafikov@innopolis.university](mailto:i.nafikov@innopolis.university)
  - **GitHub:** <https://github.com/iskanred>
- 

## CCF Lab 4 - Incident response and log management

### Task description

In this lab, you will set up a complete environment for detecting and responding to security incidents using the Wazuh security platform. You will simulate real-world attacks, configure automated responses, and explore best practices for managing logs and system alerts. This hands-on exercise is designed to reinforce key concepts in incident detection, active response, and log retention.

## Task 1 - Preparation

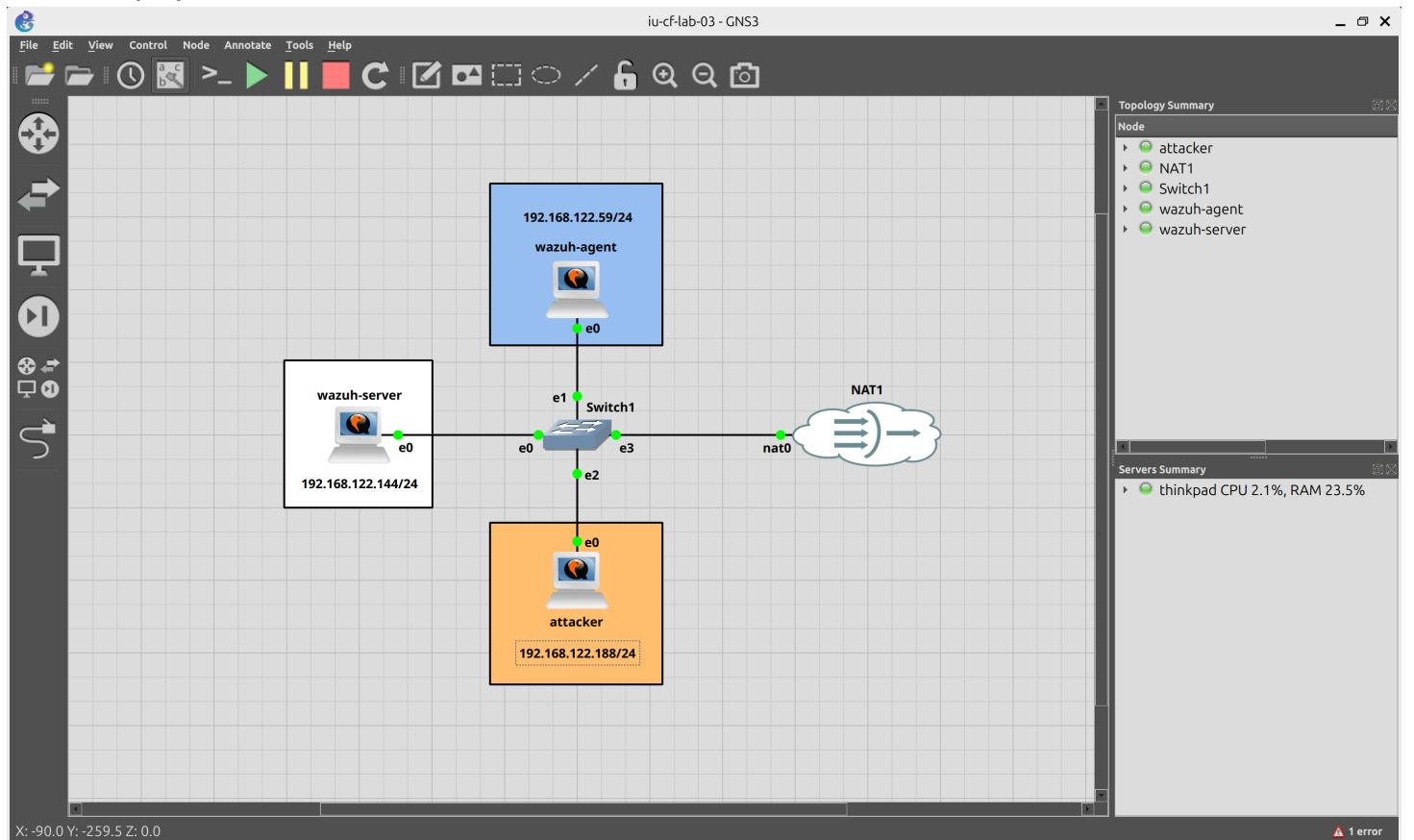
---

### 1.

### Task description

Deploy Wazuh central components (server, indexer, and dashboard) on your host machine or use an alternate method like Docker.

- First, I deployed 3 VMs inside the GNS3



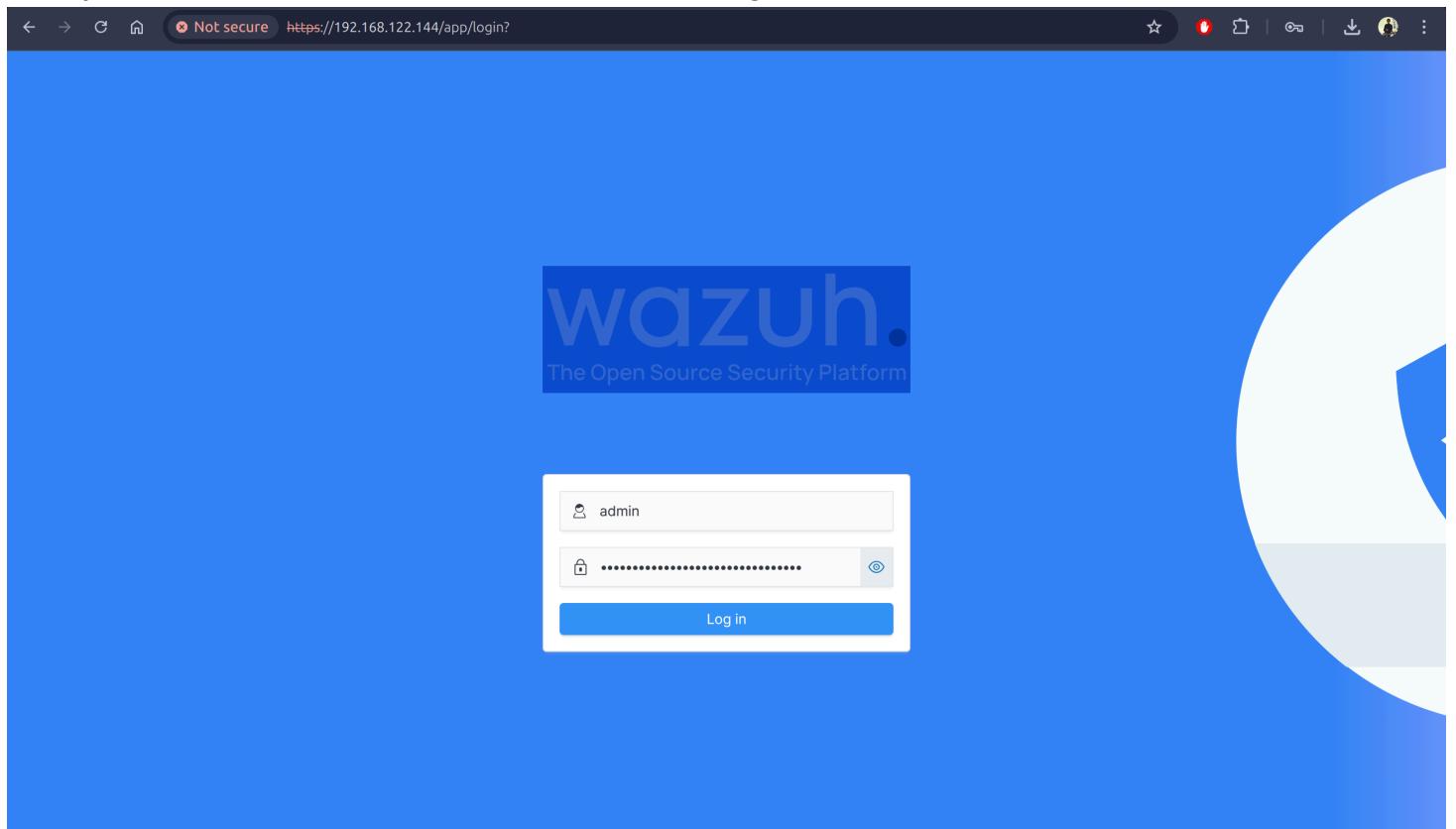
- I entered the Wazuh server (manager + indexer + dashboard) VM and exported \$TYPE environment variable

```
Terminal      x   Terminal      x   ubuntu@ubuntu-cloud:~ x   ubuntu@ubuntu-cloud:~ x   ubuntu@ubuntu-cloud:~ x
ubuntu@ubuntu-cloud:~$ echo $TYPE
wazuh-server
ubuntu@ubuntu-cloud:~$
```

- Then I installed the Wazuh following the [Quickstart](#) instruction from the official website

```
Terminal      x   Terminal      x   ubuntu@ubuntu-cloud:~ x   ubuntu@ubuntu-cloud:~ x   ubuntu@ubuntu-cloud:~ x   Terminal      x
ubuntu@ubuntu-cloud:~$ curl -s0 https://packages.wazuh.com/4.11/wazuh-install.sh && sudo bash ./wazuh-install.sh -a
27/04/2025 17:20:50 INFO: Starting Wazuh installation assistant. Wazuh version: 4.11.2 (x86_64/AMD64)
27/04/2025 17:20:50 INFO: Verbose logging redirected to /var/log/wazuh-install.log
27/04/2025 17:20:50 INFO: The recommended systems are: Red Hat Enterprise Linux 7, 8, 9; CentOS 7, 8; Amazon Linux 2; Ubuntu 16.0
4, 18.04, 20.04, 22.04.
27/04/2025 17:20:50 WARNING: The current system does not match with the list of recommended systems. The installation may not wor
k properly.
27/04/2025 17:21:13 INFO: Verifying that your system meets the recommended minimum hardware requirements.
27/04/2025 17:21:13 INFO: Wazuh web interface port will be 443.
27/04/2025 17:21:20 INFO: --- Dependencies ---
27/04/2025 17:21:20 INFO: Installing apt-transport-https.
27/04/2025 17:21:24 INFO: Installing debhelper.
27/04/2025 17:22:11 INFO: Wazuh repository added.
27/04/2025 17:22:11 INFO: --- Configuration files ---
27/04/2025 17:22:11 INFO: Generating configuration files.
27/04/2025 17:22:11 INFO: Generating the root certificate.
27/04/2025 17:22:12 INFO: Generating Admin certificates.
27/04/2025 17:22:12 INFO: Generating Wazuh indexer certificates.
27/04/2025 17:22:12 INFO: Generating Filebeat certificates.
27/04/2025 17:22:12 INFO: Generating Wazuh dashboard certificates.
27/04/2025 17:22:12 INFO: Created wazuh-install-files.tar. It contains the Wazuh cluster key, certificates, and passwords necessa
ry for installation.
27/04/2025 17:22:13 INFO: --- Wazuh indexer ---
27/04/2025 17:22:13 INFO: Starting Wazuh indexer installation.
27/04/2025 17:26:12 INFO: Wazuh indexer installation finished.
27/04/2025 17:26:12 INFO: Wazuh indexer post-install configuration finished.
27/04/2025 17:26:12 INFO: Starting service wazuh-indexer.
27/04/2025 17:26:27 INFO: wazuh-indexer service started.
27/04/2025 17:26:27 INFO: Initializing Wazuh indexer cluster security settings.
27/04/2025 17:26:31 INFO: Wazuh indexer cluster security configuration initialized.
27/04/2025 17:26:31 INFO: Wazuh indexer cluster initialized.
27/04/2025 17:26:31 INFO: --- Wazuh server ---
27/04/2025 17:26:31 INFO: Starting the Wazuh manager installation.
27/04/2025 17:29:54 INFO: Wazuh manager installation finished.
```

- Finally, I was able to access Wazuh dashboard through HTTPS



- After logging in with generated password I went to the Dashboard's

**LAST 24 HOURS ALERTS**

Critical severity	High severity	Medium severity	Low severity
0	0	18	11
Rule level 15 or higher	Rule level 12 to 14	Rule level 7 to 11	Rule level 0 to 6

**AGENTS SUMMARY**

This instance has no agents registered.  
Please deploy agents to begin monitoring your endpoints.

**ENDPOINT SECURITY**

- Configuration Assessment**: Scan your assets as part of a configuration assessment audit.
- Malware Detection**: Check indicators of compromise triggered by malware infections or cyberattacks.
- File Integrity Monitoring**: Alerts related to file changes, including permissions, content, ownership, and attributes.

**THREAT INTELLIGENCE**

- Threat Hunting**: Browse through your security alerts, identifying issues and threats in your environment.
- Vulnerability Detection**: Discover what applications in your environment are affected by well-known vulnerabilities.
- MITRE ATT&CK**: Explore security alerts mapped to adversary tactics and techniques for better threat understanding.

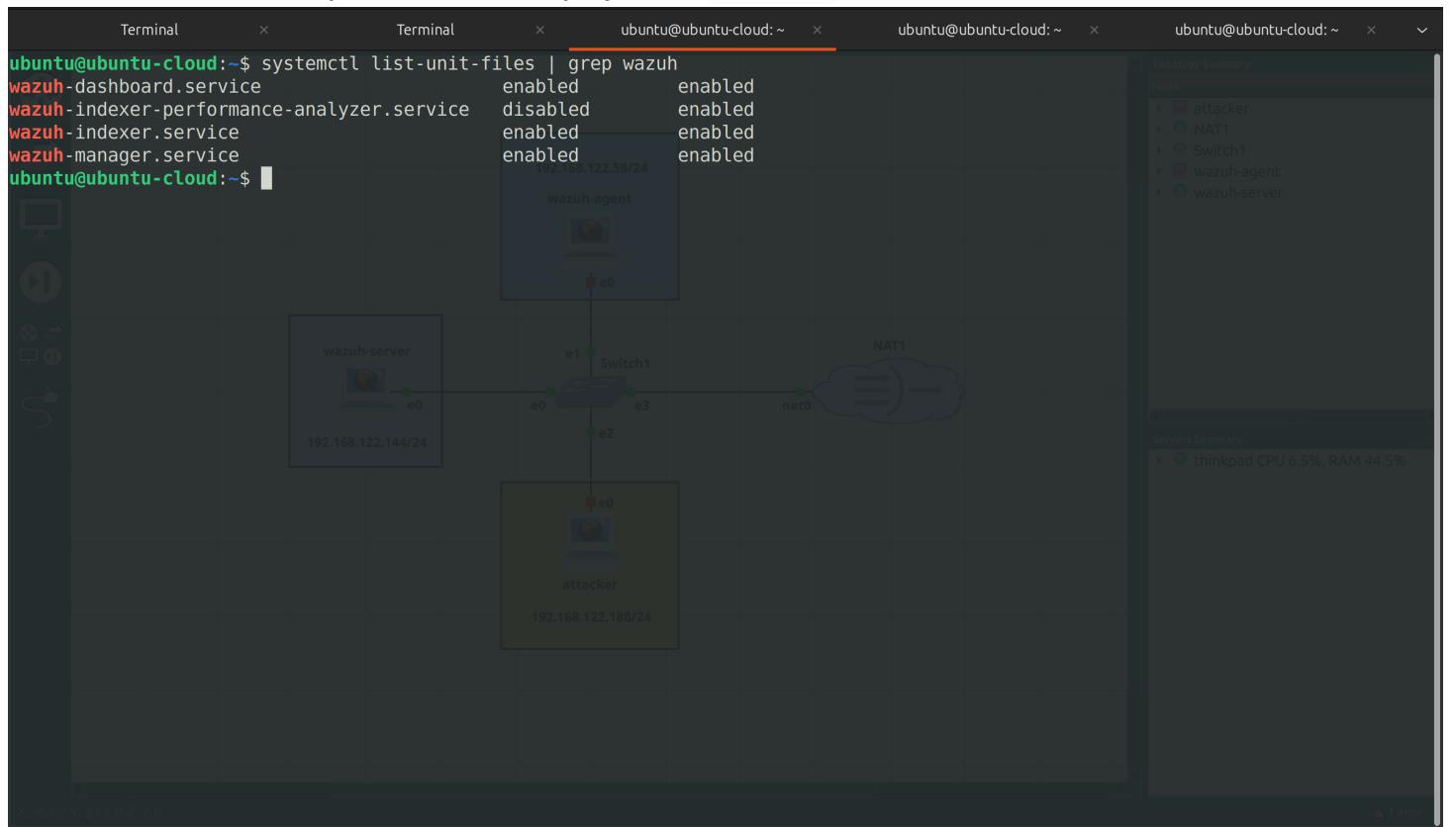
**SECURITY OPERATIONS**

- PCI DSS**
- GDPR**

**CLOUD SECURITY**

- Docker**
- AWS Amazon Web Services**

- As we see Wazuh components were deployed as systemd services

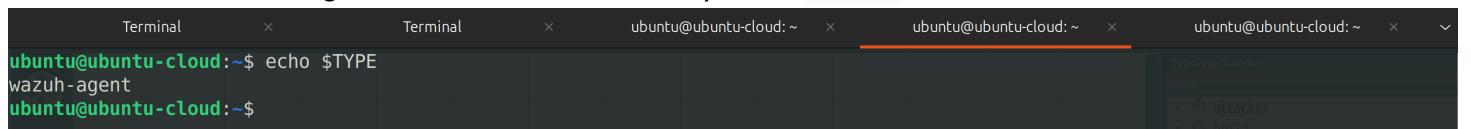


## 2.

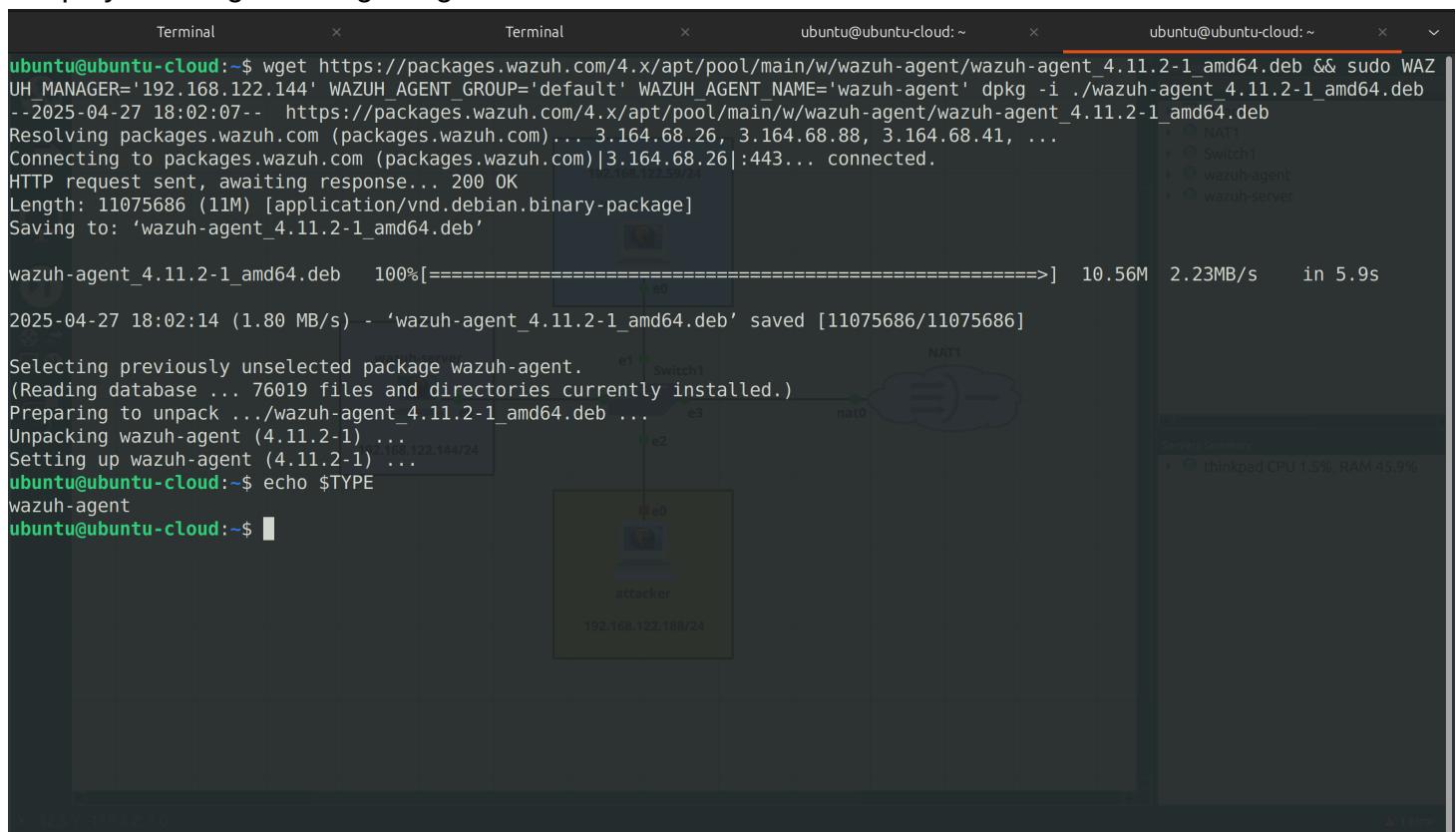
### Task description

Set up a second machine with a Unix-like OS, install the Wazuh agent, and enroll it with the Wazuh manager. Enable SSH and create a test user account.

- I entered the Wazuh agent for Ubuntu VM and exported \$TYPE environment variable



- I deployed an agent using the generated command on the Wazuh Dashboard



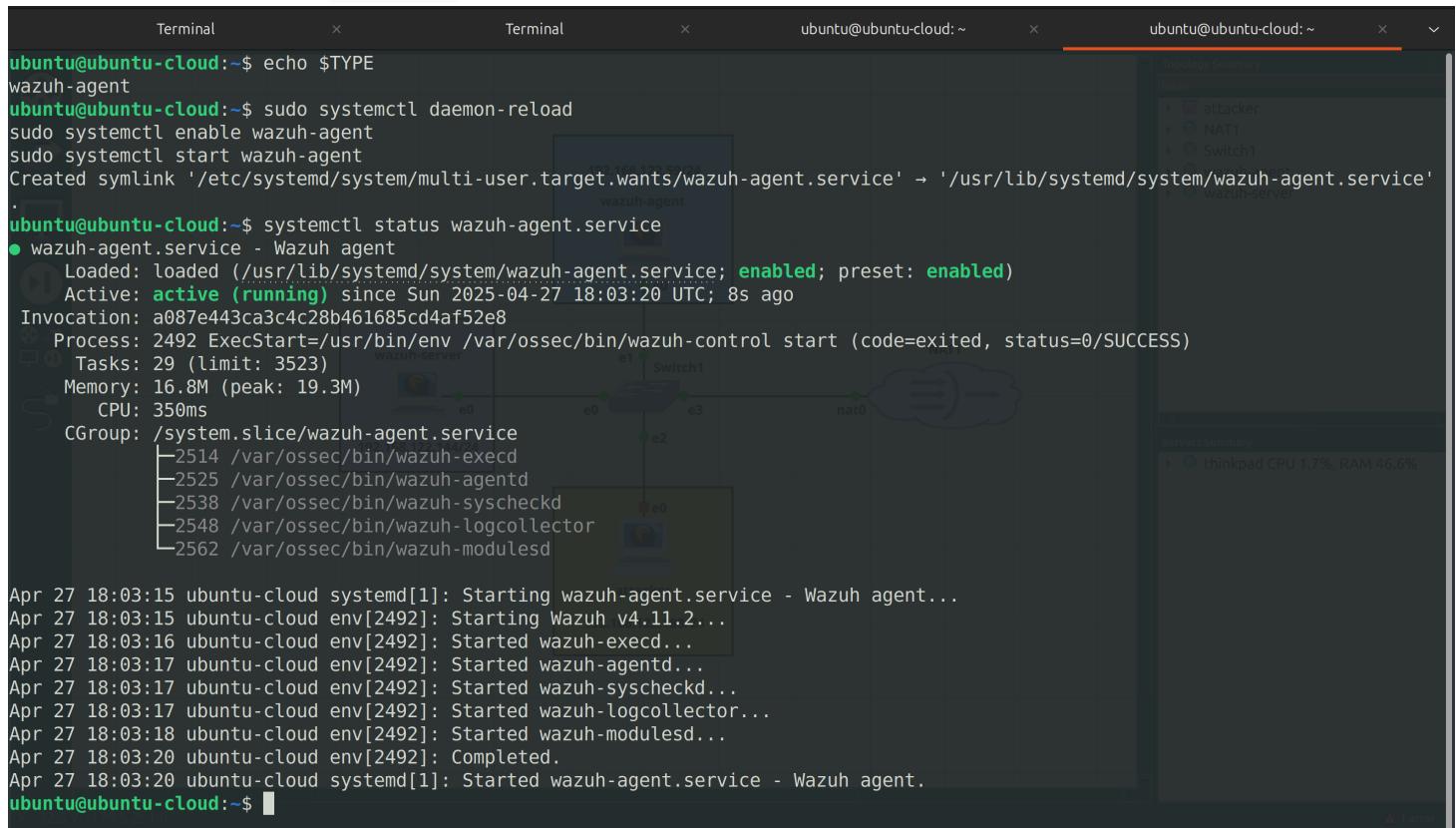
```
ubuntu@ubuntu-cloud:~$ wget https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.11.2-1_amd64.deb && sudo dpkg -i ./wazuh-agent_4.11.2-1_amd64.deb
UH_MANAGER='192.168.122.144' WAZUH_AGENT_GROUP='default' WAZUH_AGENT_NAME='wazuh-agent' dpkg -i ./wazuh-agent_4.11.2-1_amd64.deb
--2025-04-27 18:02:07-- https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.11.2-1_amd64.deb
Resolving packages.wazuh.com (packages.wazuh.com)... 3.164.68.26, 3.164.68.88, 3.164.68.41, ...
Connecting to packages.wazuh.com (packages.wazuh.com)|3.164.68.26|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11075686 (11M) [application/vnd.debian.binary-package]
Saving to: 'wazuh-agent_4.11.2-1_amd64.deb'

wazuh-agent_4.11.2-1_amd64.deb    100%[=====] 10.56M 2.23MB/s   in 5.9s

2025-04-27 18:02:14 (1.80 MB/s) - 'wazuh-agent_4.11.2-1_amd64.deb' saved [11075686/11075686]

Selecting previously unselected package wazuh-agent.
(Reading database ... 76019 files and directories currently installed.)
Preparing to unpack .../wazuh-agent_4.11.2-1_amd64.deb ...
Unpacking wazuh-agent (4.11.2-1) ...
Setting up wazuh-agent (4.11.2-1) ...
ubuntu@ubuntu-cloud:~$ echo $TYPE
wazuh-agent
ubuntu@ubuntu-cloud:~$
```

- Finally, I launched the `systemd` service of the Wazuh Agent



```
ubuntu@ubuntu-cloud:~$ echo $TYPE
wazuh-agent
ubuntu@ubuntu-cloud:~$ sudo systemctl daemon-reload
sudo systemctl enable wazuh-agent
sudo systemctl start wazuh-agent
Created symlink '/etc/systemd/system/multi-user.target.wants/wazuh-agent.service' → '/usr/lib/systemd/system/wazuh-agent.service'.
.
ubuntu@ubuntu-cloud:~$ systemctl status wazuh-agent.service
● wazuh-agent.service - Wazuh agent
   Loaded: loaded (/usr/lib/systemd/system/wazuh-agent.service; enabled; preset: enabled)
   Active: active (running) since Sun 2025-04-27 18:03:20 UTC; 8s ago
     Invocation: a087e443ca3c4c28b461685cd4af52e8
   Process: 2492 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (code=exited, status=0/SUCCESS)
   Tasks: 29 (limit: 3523)
   Memory: 16.8M (peak: 19.3M)
      CPU: 350ms
     CGroup: /system.slice/wazuh-agent.service
             ├─2514 /var/ossec/bin/wazuh-execd
             ├─2525 /var/ossec/bin/wazuh-agentd
             ├─2538 /var/ossec/bin/wazuh-syscheckd
             ├─2548 /var/ossec/bin/wazuh-logcollector
             └─2562 /var/ossec/bin/wazuh-modulesd

Apr 27 18:03:15 ubuntu-cloud systemd[1]: Starting wazuh-agent.service - Wazuh agent...
Apr 27 18:03:15 ubuntu-cloud env[2492]: Starting Wazuh v4.11.2...
Apr 27 18:03:16 ubuntu-cloud env[2492]: Started wazuh-execd...
Apr 27 18:03:17 ubuntu-cloud env[2492]: Started wazuh-agentd...
Apr 27 18:03:17 ubuntu-cloud env[2492]: Started wazuh-syscheckd...
Apr 27 18:03:17 ubuntu-cloud env[2492]: Started wazuh-logcollector...
Apr 27 18:03:18 ubuntu-cloud env[2492]: Started wazuh-modulesd...
Apr 27 18:03:20 ubuntu-cloud env[2492]: Completed.
Apr 27 18:03:20 ubuntu-cloud systemd[1]: Started wazuh-agent.service - Wazuh agent.
ubuntu@ubuntu-cloud:~$
```

- After this I could see this agent on the Wazuh Dashboard

ID	Name	IP address	Group(s)	Operating system	Cluster node	Version	Status	Actions
001	wazuh-agent	192.168.122.59	default	Ubuntu 24.10	node01	v4.11.2	● active ⓘ	...

- Also, I could see its logs

3.

### Task description

Prepare a third machine to act as the "attacker" endpoint to simulate cyber attacks. You can collaborate with your colleagues to simulate attacks if you don't have enough hardware resources for three machines.

- I entered the attacker's Ubuntu VM and exported \$TYPE environment variable

```
Terminal      Terminal      Terminal      Terminal
ubuntu@ubuntu-cloud:~$ echo $TYPE
attacker
ubuntu@ubuntu-cloud:~$
```

The screenshot shows four terminal windows side-by-side. The first three terminals are labeled "Terminal" and the fourth is labeled "ubuntu@ubuntu-cloud: ~". In the fourth terminal, the command "echo \$TYPE" is run and the output "attacker" is displayed.

## Task 2 - Configure active response

1.

### Task description

Enable the Wazuh active response feature to disable a user account for 10 minutes when a brute force attempt is detected.

## Configuration

- Firstly, I found an [instruction](#) how to disable Linux user account in case of brute force attack on the Wazuh official website

The screenshot shows a web browser displaying the Wazuh documentation. The URL in the address bar is [documentation.wazuh.com/current/user-manual/capabilities/active-response/ar-use-cases/disabling-user-account.html](https://documentation.wazuh.com/current/user-manual/capabilities/active-response/ar-use-cases/disabling-user-account.html). The page title is "Disabling a Linux user account with Active Response". The page content explains that without knowledge of the password for an account, an adversary might opt to systematically guess the password using a repetitive or iterative mechanism. It describes how to configure the "disable-account" active response to disable a Linux/Unix account subject to brute-force attacks. The Wazuh uses section notes that it utilizes the "wazuh-authd" component to implement this functionality. The sidebar on the left shows navigation links for "User manual", "Getting started", "Quickstart", "Installation guide", and "Installation alternatives". The top navigation bar includes links for "Blog", "Community", "Contact us", social media icons, and a search bar. The top right corner shows the version "Version 4.11 (current)".

- Following the instruction I added a new alert group and a rule inside it that detects "possible password guess" for a specific user on the Wazuh Server. I made the rule triggered after 4 failed login tries in

one minute. Also, I assigned the MIT&RE attack ID = T1110 which is a [Brute Force](#).

```
<!-- Local rules -->
<!-- Modify it at your will. -->
<!-- Copyright (C) 2015, Wazuh Inc. -->

<!-- Example -->
<group name="local,syslog,sshd,>

    <!--
    Dec 10 01:02:02 host sshd[1234]: Failed none for root from 1.1.1.1 port 1066 ssh2
    -->
    <rule id="100001" level="5">
        <if_sid>5716</if_sid>
        <srcip>1.1.1.1</srcip>
        <description>sshd: authentication failed from IP 1.1.1.1.</description>
        <group>authentication_failed,pci_dss_10.2.4,pci_dss_10.2.5,</group>
    </rule>
</group>

<group name="pam,syslog,">

    <rule id="120100" level="10" frequency="4" timeframe="60">
        <if_matched_sid>5503</if_matched_sid>
        <description>Possible password guess on ${dstuser}: 4 failed logins in a short period of time</description>
        <mitre>
            <id>T1110</id>
        </mitre>
    </rule>
</group>
~

"/var/ossec/etc/rules/local_rules.xml" 31L, 807B written
```

- I made sure that there is a `disable-account` command listed inside the `/var/ossec/etc/ossec.conf`

```
<synchronization>
    <enabled>yes</enabled>
    <interval>5m</interval>
    <max_eps>10</max_eps>
</synchronization>
</syscheck>

<!-- Active response -->
<global>
    <white_list>127.0.0.1</white_list>
    <white_list>^localhost.localdomain$</white_list>
    <white_list>127.0.0.53</white_list>
</global>

<command>
    <name>disable-account</name>
    <executable>disable-account</executable>
    <timeout_allowed>yes</timeout_allowed>
</command>

<command>
    <name>restart-wazuh</name>
    <executable>restart-wazuh</executable>
</command>

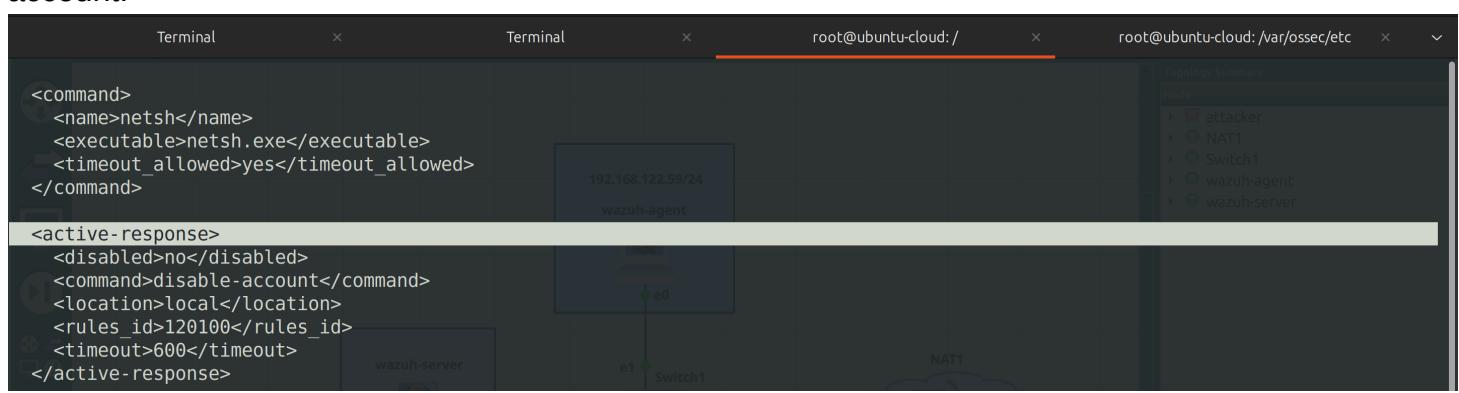
<command>
    <name>firewall-drop</name>
    <executable>firewall-drop</executable>
    <timeout_allowed>yes</timeout_allowed>
</command>

<command>
    <name>enable-wazuh</name>
    <executable>enable-wazuh</executable>
    <timeout_allowed>yes</timeout_allowed>
</command>
</active_response>

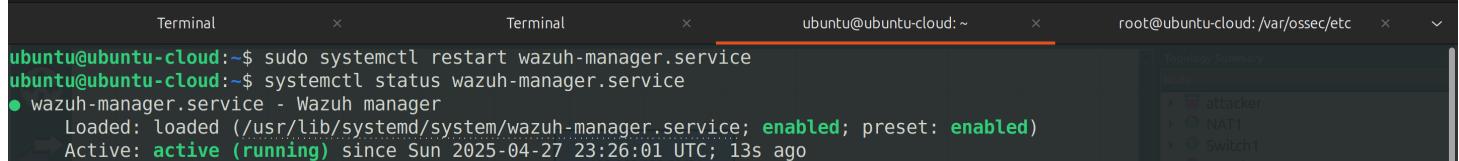
"/var/ossec/etc/ossec.conf" 328 lines --59--
```

- Then I added a new active response action inside the `/var/ossec/etc/ossec.conf`. This action executes `disable-account` command whenever the rule with ID = 120100 (which is a rule I defined to detect a brute force attack) is triggered. Also, we that this action has a required timeout for 10 minutes (600 seconds). After that period, the Active Response reverts its action and re-enables the

account.



- Finally, to apply new changes I restarted the wazuh-manager daemon

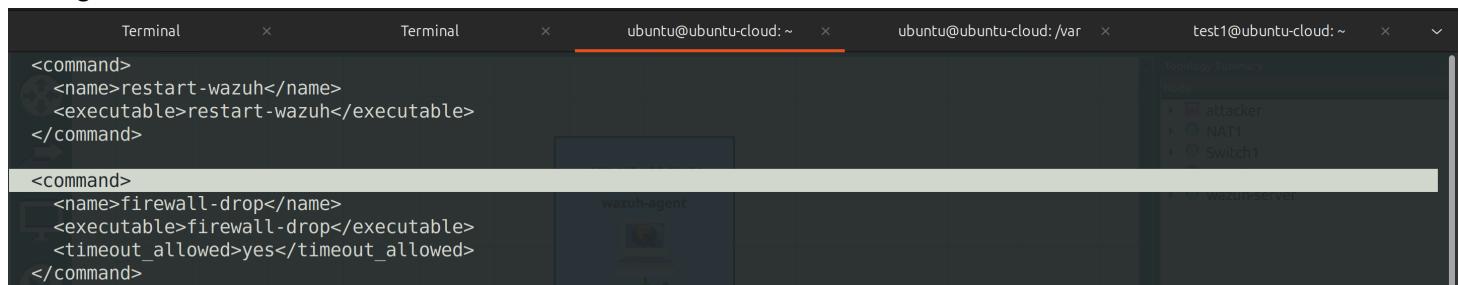


## 2.

### Task description

Also configure it to block the attacker's IP address for 10 minutes.

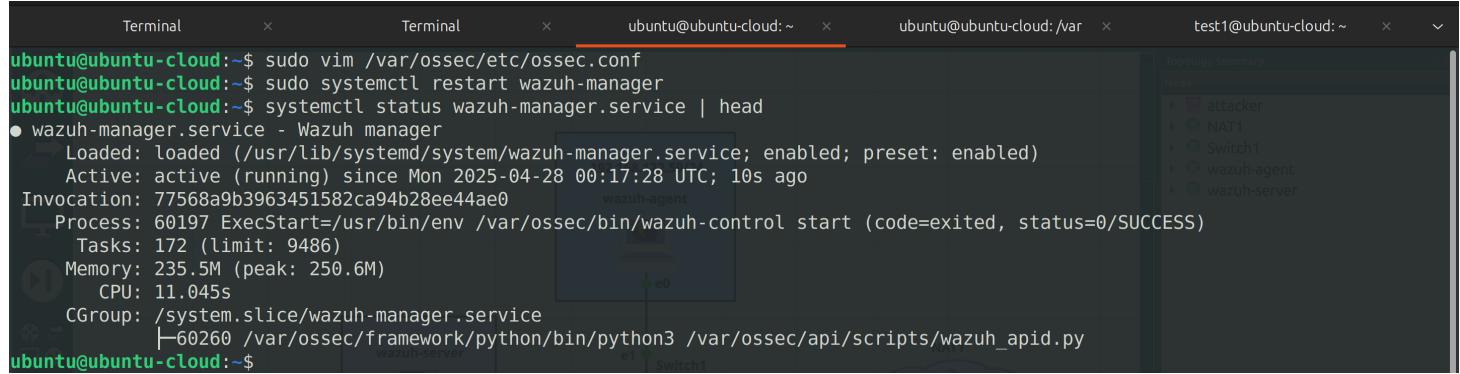
- Now I need to perform similar action. The packets from attacker's IP must be dropped for 10 minutes if the brute force is detected. I found an [instruction](#) that is pretty similar to the previous one on the Wazuh official website.
- First, let's start with making sure that the `firewall-drop` command exists inside the Wazuh Manager config



- Then I added a new active response action that drops attacker's packets if the same rule with ID = 120100, which I defined in the previous sub-task, is triggered



- Finally, I again restarted the Wazuh Manager daemon



## 3.

### Task description

Create `/home/<USERNAME>/malwarefiles/` on the monitored endpoint. Integrate malware detection (e.g., VirusTotal or YARA), and monitor this directory for malware. Configure Wazuh to automatically delete detected malware files.

- Again, I found a useful [instruction](#) to detect and delete malware files using VirusTotal on the Wazuh official website.
- First, I signed up VirusTotal to obtain an API key in order to allow Wazuh to make requests for checking a file.

virustotal.com/gui/user/iskanred/apikey

Σ Search for IoCs, Threat Actors, or Malware iskanred mains

### API KEY

This is your personal key. Do not disclose it to anyone that you do not trust, do not embed it in scripts or software from which it can be easily retrieved if you care about its confidentiality. By submitting data using your API key, you are agreeing to our [Terms of Service](#) and [Privacy Notice](#), and to the **sharing of your Sample submissions with the security community**. Please do not submit any personal information; we are not responsible for the contents of your submissions. [Learn more](#)

[Copy](#) [Open in new tab](#)

### API QUOTA ALLOWANCES FOR YOUR USER

You own a standard free end-user account. It is not tied to any corporate group and so it does not have access to Premium services. You are subjected to the following limitations:

Access level	Limited, standard free public API	<a href="#">Upgrade to premium</a>
Usage	<b>Must not be used in business workflows, commercial products or services.</b>	
Request rate	4 lookups / min	<a href="#">API reference</a>
Daily quota	500 lookups / day	<a href="#">Python client</a>
Monthly quota	15.5 K lookups / month	<a href="#">Golang library</a>

Want to learn more about how our intelligence can supercharge your security operations? check our 360 overview brief.

Want to upgrade your access? Please do not hesitate to contact us, we'll go the extra mile to make you successful.

[Go premium](#) [Use in browser](#) [Discover feeds](#) [Other services](#)

[Upgrade API](#)

## Wazuh Agent

- Then I made sure that the FIM is enabled on the `wazuh-agent` inside the `/var/ossec/etc/ossec.conf` config. 1. Wazuh FIM looks for any file addition, change, or deletion on the monitored folders. This module has the hash of these files stored and triggers alerts when it detects any changes.

```
</sca>
<!-- File integrity monitoring -->
<syscheck>
  <disabled>no</disabled>
<!-- Frequency that syscheck is executed default every 12 hours -->
<frequency>43200</frequency>
<scan_on_start>yes</scan_on_start>
<!-- Directories to check (perform all possible verifications) -->
<directories>/etc,/usr/bin,/usr/sbin</directories>
<directories>/bin,/sbin,/boot</directories>
<!-- Files/directories to ignore -->
<ignore>/etc/mtab</ignore>
<ignore>/etc/hosts.deny</ignore>
<ignore>/etc/mail/statistics</ignore>
<ignore>/etc/random-seed</ignore>
<ignore>/etc/random.seed</ignore>
<ignore>/etc/adjtime</ignore>
<ignore>/etc/httpd/logs</ignore>
<ignore>/etc/utmpx</ignore>
<ignore>/etc/wtmpx</ignore>
<ignore>/etc/cups/certs</ignore>
<ignore>/etc/dumpdates</ignore>
<ignore>/etc/svc/volatile</ignore>
<!-- File types to ignore -->
<ignore type="sregex".log$|.swp$</ignore>
```

- Then I also added /home/ubuntu/malwarefiles to be monitored in near real-time.

```

<!-- File integrity monitoring -->
<syscheck>
  <disabled>no</disabled>

  <!-- Frequency that syscheck is executed default every 12 hours -->
  <frequency>43200</frequency>

  <scan_on_start>yes</scan_on_start>

  <!-- Directories to check (perform all possible verifications) -->
  <directories>/etc,/usr/bin,/usr/sbin</directories>
  <directories>/bin,/sbin,/boot</directories>

  <!-- ADDED NOW FOR VIRUSTOTAL INTEGRATION -->
  <directories realtime="yes">/home/ubuntu/malwarefiles</directories>

  <!-- Files/directories to ignore -->
  <ignore>/etc/mtab</ignore>
  <ignore>/etc/hosts.deny</ignore>
  <ignore>/etc/mail/statistics</ignore>
  <ignore>/etc/random-seed</ignore>
  <ignore>/etc/random.seed</ignore>
  <ignore>/etc/adjtime</ignore>
  <ignore>/etc/httpd/logs</ignore>
  <ignore>/etc/utmpx</ignore>
  <ignore>/etc/wtmpx</ignore>
  <ignore>/etc/cups/certs</ignore>
  <ignore>/etc/dumpdates</ignore>
  <ignore>/etc/svc/volatile</ignore>

  <!-- File types to ignore -->

```

- Afterwards, I installed jq tool which will help us to read a log easily in order to make proper active response actions

```

ubuntu@ubuntu-cloud:~$ apt show jq | head
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Package: jq
Version: 1.7.1-3build1
Priority: optional
Section: utils
Origin: Ubuntu
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: ChangZhuo Chen (陳昌倬) <czchen@debian.org>
Bugs: https://bugs.launchpad.net/ubuntu/+filebug
Installed-Size: 115 kB
Depends: libc6 (>= 2.38), libjq1 (= 1.7.1-3build1)
ubuntu@ubuntu-cloud:~$ 

```

- Then I copied an active response script that removes malware file.

```

#!/bin/bash

LOCAL=`dirname $0`;
cd $LOCAL
cd ..

PWD=`pwd`

read INPUT_JSON
FILENAME=$(echo $INPUT_JSON | jq -r .parameters.alert.data.virustotal.source.file)
COMMAND=$(echo $INPUT_JSON | jq -r .command)
LOG_FILE="${PWD}/../logs/active-responses.log"

#----- Analyze command -----
if [ ${COMMAND} = "add" ]
then
  # Send control message to execd
  printf '{"version":1,"origin":{"name":"remove-threat","module":"active-response"},"command":"check_keys", "parameters":{"keys":[]}}\n'

  read RESPONSE
  COMMAND2=$(echo $RESPONSE | jq -r .command)
  if [ ${COMMAND2} != "continue" ]
  then
    echo "`date '+%Y/%m/%d %H:%M:%S'` $0: $INPUT_JSON Remove threat active response aborted" >> ${LOG_FILE}
    exit 0;
  fi
fi

# Removing file
rm -f $FILENAME
if [ $? -eq 0 ]; then
  "/var/ossec/active-response/bin/remove-threat.sh" [New] 37L, 1003B written

```

1,7 Top

- Moving further, I changed the permissions and file ownership for this script in order to allow only Wazuh Agent to run it.

```

ubuntu@ubuntu-cloud:~$ sudo vim /var/ossec/active-responses
ubuntu@ubuntu-cloud:~$ sudo vim /var/ossec/active-response
ubuntu@ubuntu-cloud:~$ sudo vim /var/ossec/active-response/bin/remove-threat.sh
ubuntu@ubuntu-cloud:~$ sudo chmod 750 /var/ossec/active-response/bin/remove-threat.sh
ubuntu@ubuntu-cloud:~$ sudo chown root:wazuh /var/ossec/active-response/bin/remove-threat.sh
ubuntu@ubuntu-cloud:~$ sudo ls -l /var/ossec/active-response/bin/remove-threat.sh
-rwxr-x--- 1 root wazuh 1003 Apr 28 01:28 /var/ossec/active-response/bin/remove-threat.sh
ubuntu@ubuntu-cloud:~$ 

```

- Finally, I restarted the Wazuh Agent daemon on my wazuh-agent endpoint

```

ubuntu@ubuntu-cloud:~$ sudo systemctl restart wazuh-agent
ubuntu@ubuntu-cloud:~$ systemctl status wazuh-agent
● wazuh-agent.service - Wazuh agent
   Loaded: loaded (/usr/lib/systemd/system/wazuh-agent.service; enabled; preset: enabled)
     Active: active (running) since Mon 2025-04-28 01:31:11 UTC; 4s ago
   Invocation: 35e423ac373b403cb401d5db3ee82a15
     Process: 7014 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (code=exited, status=0/SUCCESS)
       Tasks: 33 (limit: 3523)
      Memory: 18.2M (peak: 20.2M)
        CPU: 3.332s
      CGroup: /system.slice/wazuh-agent.service
              ├─7036 /var/ossec/bin/wazuh-execd
              ├─7047 /var/ossec/bin/wazuh-agentd
              ├─7061 /var/ossec/bin/wazuh-syscheckd
              ├─7072 /var/ossec/bin/wazuh-logcollector
              └─7097 /var/ossec/bin/wazuh-modulesd

Apr 28 01:31:05 ubuntu-cloud systemd[1]: Starting wazuh-agent.service - Wazuh agent...
Apr 28 01:31:05 ubuntu-cloud env[7014]: Starting Wazuh v4.11.2...
Apr 28 01:31:06 ubuntu-cloud env[7014]: Started wazuh-execd...
Apr 28 01:31:07 ubuntu-cloud env[7014]: Started wazuh-agentd...
Apr 28 01:31:07 ubuntu-cloud env[7014]: Started wazuh-syscheckd...
Apr 28 01:31:08 ubuntu-cloud env[7014]: Started wazuh-logcollector...
Apr 28 01:31:09 ubuntu-cloud env[7014]: Started wazuh-modulesd...
Apr 28 01:31:11 ubuntu-cloud env[7014]: Completed.
Apr 28 01:31:11 ubuntu-cloud systemd[1]: Started wazuh-agent.service - Wazuh agent.
ubuntu@ubuntu-cloud:~$ 

```

## Wazuh Server

- I added two new rules that are triggered whenever a file was added or modified inside the /home/ubuntu/malwarefiles directory

```

<group>authentication_failed,pci_dss_10.2.4,pci_dss_10.2.5,</group>
</rule>

</group>

<group name="pam,syslog,">
  <rule id="120100" level="10" frequency="4" timeframe="60">
    <if_matched_sid>5503</if_matched_sid>
    <description>Possible password guess on ${dstuser}: 4 failed logins in a short period of time</description>
    <mitre>
      <id>T1110</id>
    </mitre>
  </rule>
</group>

<group name="syscheck,pci_dss_11.5,nist_800_53_SI.7,>
  <rule id="100200" level="7">
    <if_sid>550</if_sid>
    <field name="file">/home/ubuntu/malwarefiles</field>
    <description>File modified in /home/ubuntu/malwarefiles directory.</description>
  </rule>
  <rule id="100201" level="7">
    <if_sid>554</if_sid>
    <field name="file">/home/ubuntu/malwarefiles</field>
    <description>File added to /home/ubuntu/malwarefiles directory.</description>
  </rule>
</group>

```

"var/ossec/etc/rules/local\_rules.xml" 47L, 1292B written

42, 28 Bot

- These rules are triggered when FIM basic rules are triggered

ID	Description	Groups	Regulatory compliance	Level	File	Path
550	Integrity checksum changed.	syscheck, syscheck_entry_m odified, syscheck_file, ossec	PCI_DSS NIST_800_53 GPG13 TSC HIPAA MITRE	7	0015-ossec_rules.xml	ruleset/rules
554	File added to the system.	syscheck, syscheck_entry_a dded, syscheck_file, ossec	PCI_DSS NIST_800_53 GPG13 TSC HIPAA GDPR	5	0015-ossec_rules.xml	ruleset/rules

From here you can manage your rules.

Manage rules files   Add new rules file   Refresh   Export formatted

WQL   Custom rules

Rows per page: 10

- Then I added VirusTotal integration inside the Wazuh Manager config. My API Key is masked. This allows to trigger a VirusTotal query whenever any of the rules 100200 and 100201 are triggered.

```

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

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

  <localfile>
    <log_format>syslog</log_format>
    <location>/var/log/dpkg.log</location>
  </localfile>
</ossec_config>
<!-- VIRUSTOTAL INTEGRATION CONFIG -->
<ossec_config>
  <integration>
    <name>virustotal</name>
    <api_key>*****</api_key>
    <rule_id>100200,100201</rule_id>
    <alert_format>json</alert_format>
  </integration>
</ossec_config>
"/var/ossec/etc/ossec.conf" 352L, 9769B written
  
```

- Also, I added a remove-threat command that executes my remove-threat.sh script created above and remove-thread active response action that uses this command whenever a rule with ID = 87105 is triggered.

```

<localfile>
  <log_format>syslog</log_format>
  <location>/var/log/dpkg.log</location>
</localfile>
</ossec_config>
<!-- VIRUSTOTAL INTEGRATION CONFIG -->
<ossec_config>
  <integration>
    <name>virustotal</name>
    <api_key>d2fbab658d8bf307bc435d8447f9eab72c99b7053db68fa504d78c6830a5eb8</api_key>
    <rule_id>100200,100201</rule_id> <!-- FILE INSIDE 'malwarefiles' MODIFIED OR ADDED -->
    <alert_format>json</alert_format>
  </integration>
</ossec_config>
<!-- ACTIVE RESPONSE -->
<command>
  <name>remove-threat</name>
  <executable>remove-threat.sh</executable>
  <timeout_allowed>no</timeout_allowed> <!-- NO REVERT AFTER TIME -->
</command>

<active-response>
  <disabled>no</disabled>
  <command>remove-threat</command>
  <location>local</location>
  <rules_id>87105</rules_id> <!-- MALWARE DETECTED -->
</active-response>
"/var/ossec/etc/ossec.conf" 366L, 10258B written
  
```

- The rule 87105 is triggered when VirusTotal detects a malware inside the file sent to it.

Not secure https://192.168.122.144/app/rules#/manager/?tab=ruleset

W. Rules

Rules (1)

From here you can manage your rules.

ID ↑	Description	Groups	Regulatory compliance	Level	File	Path
87105	VirusTotal: Alert - <a href="#">virustotal.source.file</a> - <a href="#">virustotal.positives</a> engines detected this file	virustotal	PCI_DSS, GDPR, MITRE	12	<a href="#">0490-virustotal_rules.xml</a>	ruleset/rules

Rows per page: 10 < 1 >

- Then I added two more rules that says if removing a malware file was successful or not

```

<!-- FILE MODIFICATION IN 'malwarefiles' -->
<group name="syscheck,pci_dss_11.5,nist_800_53_SI.7,">

  <rule id="100200" level="7">
    <if_sid>550</if_sid>
    <field name="file">/home/ubuntu/malwarefiles</field>
    <description>File modified in /home/ubuntu/malwarefiles directory.</description>
  </rule>

  <rule id="100201" level="7">
    <if_sid>554</if_sid>
    <field name="file">/home/ubuntu/malwarefiles</field>
    <description>File added to /home/ubuntu/malwarefiles directory.</description>
  </rule>
</group>

<!-- REMOVING MALWARE FILE -->
<group name="virustotal,">
  <rule id="100092" level="12">
    <if_sid>657</if_sid>
    <match>Successfully removed threat</match>
    <description>$(parameters.program) removed threat located at $(parameters.alert.data.virustotal.source.file)</description>
  </rule>

  <rule id="100093" level="12">
    <if_sid>657</if_sid>
    <match>Error removing threat</match>
    <description>Error removing threat located at $(parameters.alert.data.virustotal.source.file)</description>
  </rule>
</group>

```

/var/ossec/etc/rules/local\_rules.xml" 64L, 1895B written 64,8 Bot

- The rule 657 is triggered whenever an active response action is done

Not secure https://192.168.122.144/app/rules#/manager/?tab=ruleset

W. Rules

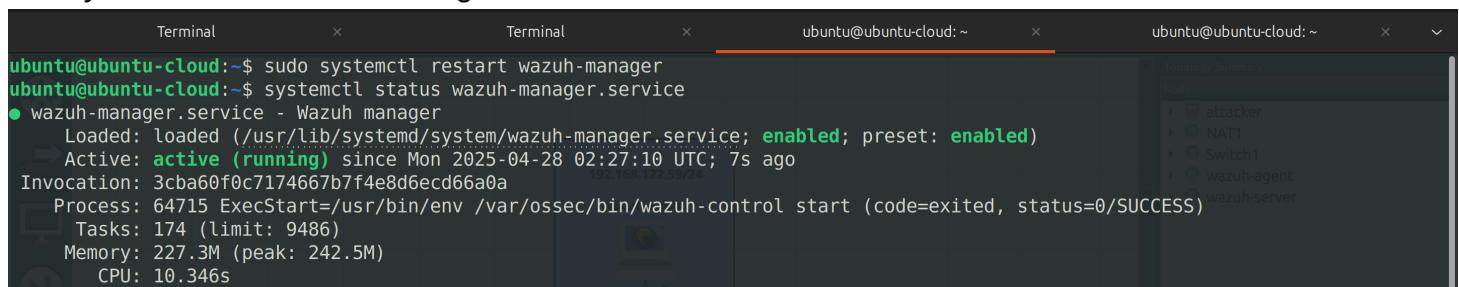
Rules (1)

From here you can manage your rules.

ID ↑	Description	Groups	Regulatory compliance	Level	File	Path
657	Active response: <a href="#">parameters.program - command</a>	active_response, ossec	PCI_DSS, GDPR, NIST_800_53, TSC	3	<a href="#">0015-ossec_rules.xml</a>	ruleset/rules

Rows per page: 10 < 1 >

- Finally, I restarted Wazuh Manager daemon



```
ubuntu@ubuntu-cloud:~$ sudo systemctl restart wazuh-manager
ubuntu@ubuntu-cloud:~$ systemctl status wazuh-manager.service
● wazuh-manager.service - Wazuh manager
    Loaded: loaded (/usr/lib/systemd/system/wazuh-manager.service; enabled; preset: enabled)
    Active: active (running) since Mon 2025-04-28 02:27:10 UTC; 7s ago
      Invocation: 3cba60f0c7174667b7f4e8d6ecd66a0a
    Process: 64715 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (code=exited, status=0/SUCCESS)
      Tasks: 174 (limit: 9486)
     Memory: 227.3M (peak: 242.5M)
        CPU: 10.346s
```

- To sum up the steps to detect, remove, and notify are the following:

1. FIM detects a file added or modified inside the `/home/ubuntu/malwarefiles`
2. The rules about file addition modification is triggered
3. Modified or added files are sent to the VirusTotal
4. After VirusTotal returns the result Wazuh checks if files are malware
5. If so, the active response is started: the files are removed.
6. If removal is successful, the rule is triggered
7. If not, another rule is triggered

## Task 3 - Simulate attacks

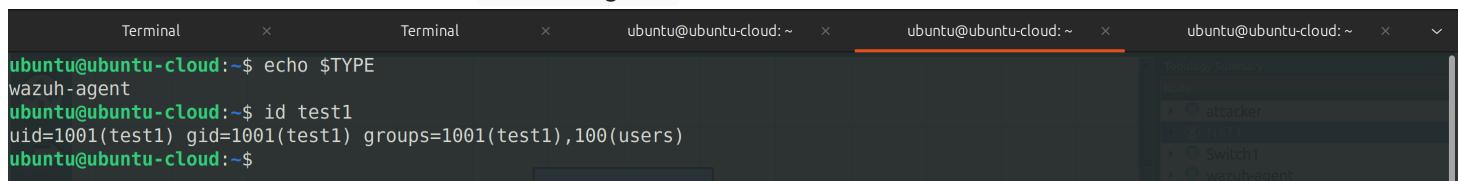
---

### 1.

#### Task description

Launch an SSH brute force attack from the attacker endpoint against the test user

- First, I created a test user on the `wazuh-agent`



```
ubuntu@ubuntu-cloud:~$ echo $TYPE
wazuh-agent
ubuntu@ubuntu-cloud:~$ id test1
uid=1001(test1) gid=1001(test1) groups=1001(test1),100(users)
ubuntu@ubuntu-cloud:~$
```

- Also, I installed [hydra](#) tool on the `attacker` machine. This tool allows to make brute force attacks easily.

```

ubuntu@ubuntu-cloud:~$ echo $TYPE
attacker
ubuntu@ubuntu-cloud:~$ apt show hydra | head
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Package: hydra
Version: 9.5-2
Priority: extra
Section: universe/net
Origin: Ubuntu
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Debian Security Tools <team+pkg-security@tracker.debian.org>
Bugs: https://bugs.launchpad.net/ubuntu/+filebug
Installed-Size: 941 kB
Depends: libapr1t64 (>= 1.2.7), libbison-1.0-0t64 (>= 1.27.5), libc6 (>= 2.38), libfbclient2 (>= 2.5.0.25784-ReleaseCandidate1.ds2), libfreerdp3-3 (>= 3.0.0), libgcrypt20 (>= 1.11.0), libidn12 (>= 1.13), libmemcached11t64 (>= 1.1.4), libmongoc-1.0-0t64 (>= 1.27.5), libmysqlclient21 (>= 8.0.11), libpcre2-8-0 (>= 10.22), libpq5, libssh-4 (>= 0.8.0), libssl3t64 (>= 3.0.0), libsvn1 (>= 1.1.0), libtinfo6 (>= 6), libwinpr3-3 (>= 3.0.0), zlib1g (>= 1:1.1.4)
ubuntu@ubuntu-cloud:~$

```

- I created 4 password to make 4 login tries through SSH

```

ubuntu@ubuntu-cloud:~$ vim passwords
ubuntu@ubuntu-cloud:~$ cat passwords
1
2
3
4
ubuntu@ubuntu-cloud:~$

```

- I started a brute force attack which was completed quickly with exactly 4 login tries

```

ubuntu@ubuntu-cloud:~$ hydra -l test1 -P passwords ssh://192.168.122.59
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-04-28 03:15:49
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 4 tasks per 1 server, overall 4 tasks, 4 login tries (l:1/p:4), ~1 try per task
[DATA] attacking ssh://192.168.122.59:22/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-04-28 03:15:52

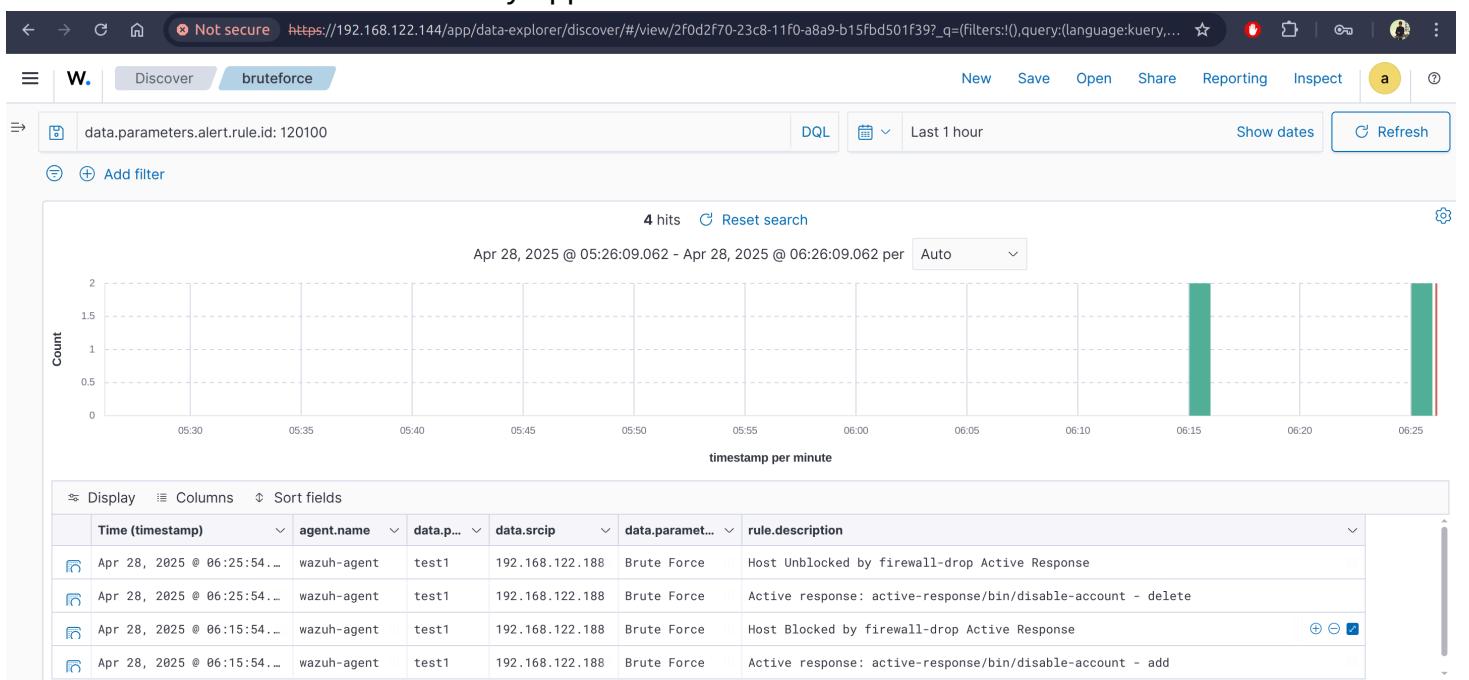
```

## 2.

### Task description

Show relevant alerts on the Wazuh dashboard.

- Below we see that the alerts actually appeared on the dashboard



- From them we can see that the MIT&RE attack type was successfully recognized (Brute Force). We see the attacker's IP address and rules description.
  - Host Blocked
  - disable-account - add
- Also, we see alerts that also notifies us about disabling applied active response actions after exactly 10 minutes
  - Host Unblocked
  - disable account - delete

### 3.

#### Task description

Provide evidence that the user account was disabled and the attacker IP blocked. Include dashboard alerts and screenshots from the endpoint involved.

- We see that the attacker's **IP address was blocked** since an attacker cannot successfully ping the wazuh-agent endpoint

```

Terminal      Terminal      ubuntu@ubuntu-cloud:~      ubuntu@ubuntu-cloud:~      ubuntu@ubuntu-cloud:~      topology summary
ubuntu@ubuntu-cloud:~$ hydra -l test1 -P passwords ssh://192.168.122.59
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-04-28 03:15:49
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 4 tasks per 1 server, overall 4 tasks, 4 login tries (l:1:p:4), ~1 try per task
[DATA] attacking ssh://192.168.122.59:22/          wazuh-agent
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-04-28 03:15:52
ubuntu@ubuntu-cloud:~$ ping 192.168.122.59
PING 192.168.122.59 (192.168.122.59) 56(84) bytes of data.

^C
--- 192.168.122.59 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2062ms
NAT1
wazuh-agent
attacker
Switch1
ubuntu@ubuntu-cloud:~$
```

- Meanwhile, ping requests were sent successfully but with no reply

No.	Time	Source	Destination	Protocol	Length	Info
5	6.434394	192.168.122.188	192.168.122.59	ICMP	98	Echo (ping) request id=0x0886, seq=1/256, ttl=64 (no response found!)
6	7.498493	192.168.122.188	192.168.122.59	ICMP	98	Echo (ping) request id=0x0886, seq=2/512, ttl=64 (no response found!)
11	8.522161	192.168.122.188	192.168.122.59	ICMP	98	Echo (ping) request id=0x0886, seq=3/768, ttl=64 (no response found!)
12	9.546458	192.168.122.188	192.168.122.59	ICMP	98	Echo (ping) request id=0x0886, seq=4/1024, ttl=64 (no response found!)
14	10.570476	192.168.122.188	192.168.122.59	ICMP	98	Echo (ping) request id=0x0886, seq=5/1280, ttl=64 (no response found!)
17	11.594478	192.168.122.188	192.168.122.59	ICMP	98	Echo (ping) request id=0x0886, seq=6/1536, ttl=64 (no response found!)

- Also, we see that the test1 user account was disabled

```

Terminal      Terminal      ubuntu@ubuntu-cloud:~      ubuntu@ubuntu-cloud:~      ubuntu@ubuntu-cloud:~      topology summary
ubuntu@ubuntu-cloud:~$ sudo passwd --status test1
test1 * 2025-04-27 0 99999 7 -1
ubuntu@ubuntu-cloud:~$
```

**-S, --status**  
Display account status information. The status information consists of 7 fields. The first field is the user's login name. The second field indicates if the user account has a locked password (L), has no password (NP), or has a usable password (P). The third field gives the date of the last password change. The next four fields are the minimum age, maximum age, warning period, and inactivity period for the password. These ages are expressed in days.

- And after 10 minutes everything works again

```

Terminal      Terminal      ubuntu@ubuntu-cloud:~      ubuntu@ubuntu-cloud:~      ubuntu@ubuntu-cloud:~      topology summary
ubuntu@ubuntu-cloud:~$ hydra -l test1 -P passwords ssh://192.168.122.59
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-04-28 03:15:49
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 4 tasks per 1 server, overall 4 tasks, 4 login tries (l:1:p:4), ~1 try per task
[DATA] attacking ssh://192.168.122.59:22/          wazuh-agent
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-04-28 03:15:52
ubuntu@ubuntu-cloud:~$ ping 192.168.122.59
PING 192.168.122.59 (192.168.122.59) 56(84) bytes of data.

^C
--- 192.168.122.59 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2062ms
NAT1
wazuh-agent
attacker
Switch1
ubuntu@ubuntu-cloud:~$
```

## 4.

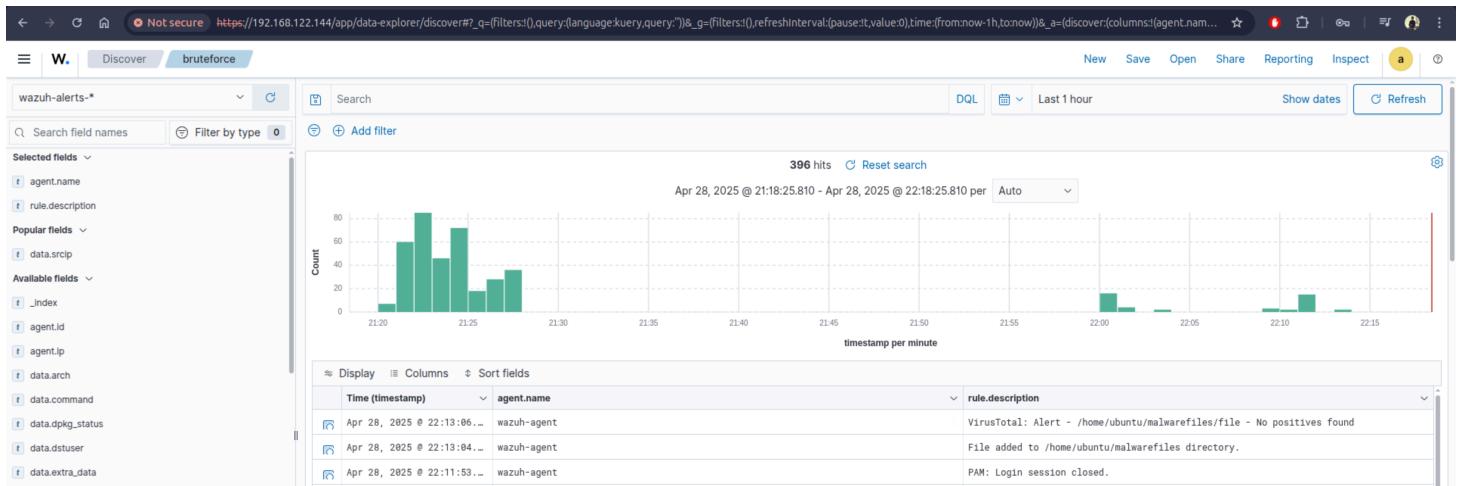
### Task description

Download a malware sample into the `/home/<USERNAME>/malwarefiles/` directory. Confirm detection and show that it was automatically removed.

- First, let me check if our rules work correctly
- I added a new file to the `/home/ubuntu/malwarefiles/` directory which is a safe empty file

```
ubuntu@ubuntu-cloud:~/malwarefiles$ touch file  
ubuntu@ubuntu-cloud:~/malwarefiles$
```

- And we can instantly see the alerts fired on Wazuh Dashboard



- We see that this file was not malware. Hence, it was not removed

```
ubuntu@ubuntu-cloud:~/malwarefiles$ ls -l  
total 0  
-rw-rw-r-- 1 ubuntu ubuntu 0 Apr 28 19:13 file
```

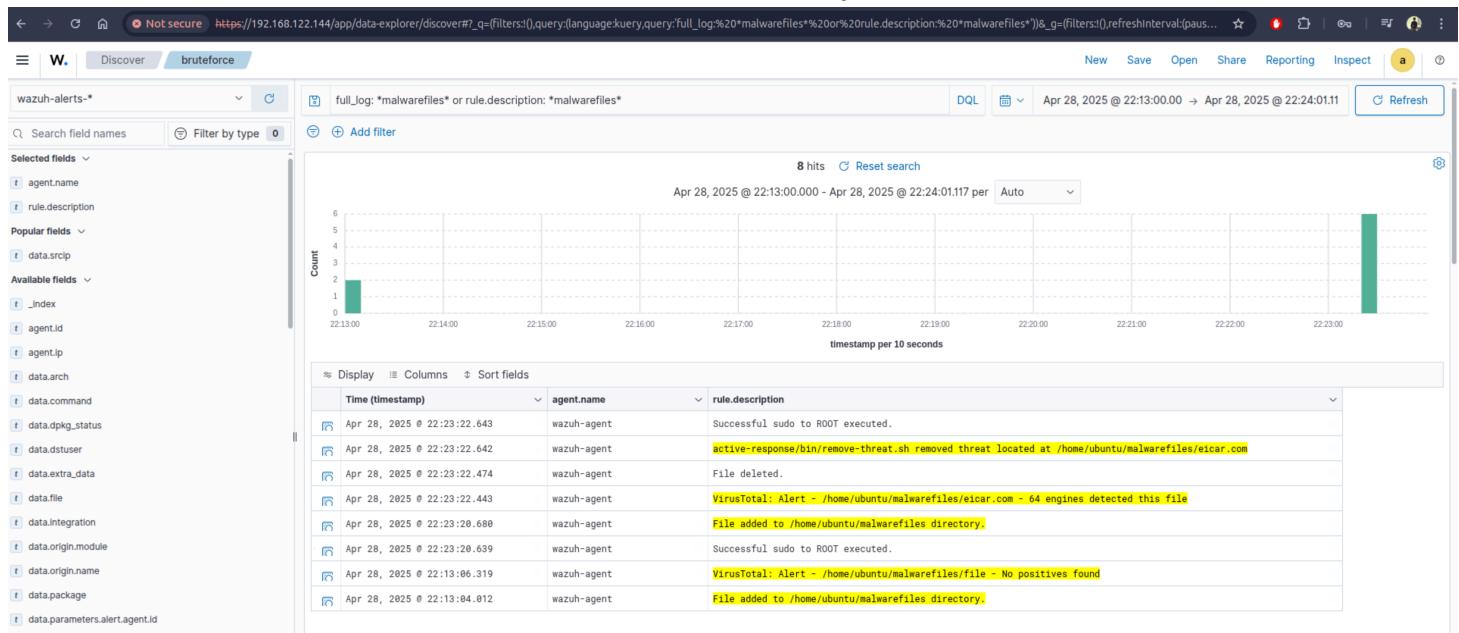
- Now let's check a malware file
- I downloaded a file that is used to check if antivirus software works: [eicar.com](http://eicar.com)

```
ubuntu@ubuntu-cloud:~/malwarefiles$ sudo curl -Lo ./eicar.com https://secure.eicar.org/eicar.com && sudo ls -lah ./eicar.com  
% Total % Received % Xferd Average Speed Time Time Current  
          Dload Upload Total Spent Left Speed  
100  68 100  68  0    0   81      0 --:--:-- --:--:-- 81  
-rw-r--r-- 1 root root 68 Apr 28 19:23 ./eicar.com  
ubuntu@ubuntu-cloud:~/malwarefiles$
```

- As we see the file was actually created
- However, it had been quickly deleted

```
ubuntu@ubuntu-cloud:~/malwarefiles$ ls -l  
total 0  
-rw-rw-r-- 1 ubuntu ubuntu 0 Apr 28 19:13 file  
ubuntu@ubuntu-cloud:~/malwarefiles$
```

- And on the dashboard we can see that the files was actually sent to VirusTotal and deleted after



## Task 4 - Log management

### 1.

#### Task description

Define what a log retention policy is.

- A **log retention policy** is a defined set of guidelines that specifies how long log records should be retained, stored, and eventually deleted. This policy typically considers factors such as:
  - Compliance requirements: Meeting legal or regulatory mandates concerning data retention.
  - Storage limitations: Managing disk space and storage costs by determining how long logs should be kept.
  - Operational needs: Establishing a timeframe for which logs are useful for troubleshooting, auditing, or performance monitoring.
- A well-defined log retention policy helps organizations balance the need for retaining logs for analysis and compliance against the costs and risks associated with storing large volumes of data.

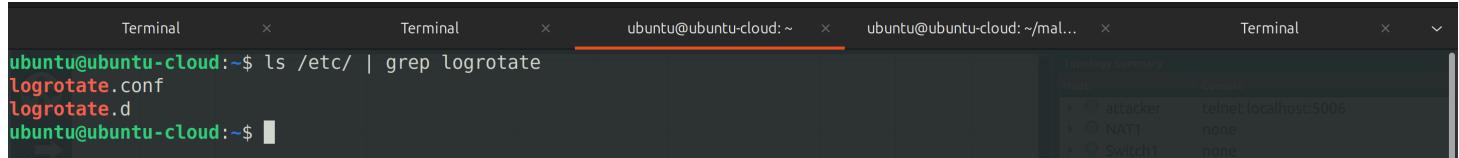
### 2.

#### Task description

Explain how logs are rotated on Linux and how disk space is managed in relation to logs.

# How logs are rotated

- **Log rotation** is the process of renaming, archiving, and managing log files to prevent them from consuming excessive disk space while maintaining a manageable history of log entries for analysis and auditing. In Linux, log rotation is typically handled by tools and services such as `logrotate`. It is configured through a set of configuration files (commonly located in `/etc/logrotate.conf` and individual configuration files in `/etc/logrotate.d/` ).



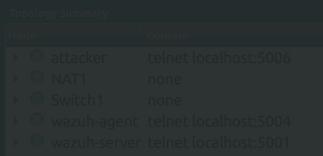
```
Terminal      Terminal      Terminal      Terminal      Terminal
ubuntu@ubuntu-cloud:~$ ls /etc/ | grep logrotate
logrotate.conf
logrotate.d
ubuntu@ubuntu-cloud:~$
```

- `logrotate` allows customization of rotation policies, including:
  - **Frequency**: Logs can be rotated daily, weekly, monthly, or based on size.
  - **Retention**: Determine how many old log files to keep (e.g., keep the last 4 rotated logs).
  - **Compression**: Old logs can be compressed (e.g., using gzip) to save disk space.
  - **Post-rotation actions**: Commands can be specified to run after rotation, such as restarting services (e.g., systemd services or rsyslog).
- Example of a `logrotate` configuration:

```
/var/log/example.log {
    daily                      # Rotate daily
    missingok                  # Don't throw an error if the log file is missing
    rotate 7                   # Keep the last 7 logs
    compress                   # Compress old logs
    delaycompress              # Delay compression until the next rotation
    notifempty                 # Do not rotate the log if it is empty
    create 0640 root adm      # Create a new log file with given permissions and
                               # ownership
    sharedscripts               # Run post-rotate scripts only once for multiple logs
    postrotate
        systemctl reload example.service
    endscript
}
```

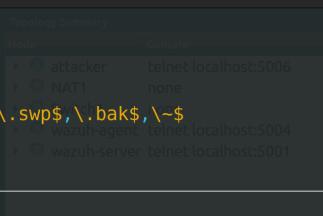
- Example of a logrotate configuration from the wazuh-server machine

```
# see "man logrotate" for details
# global options do not affect preceding include directives
# rotate log files weekly
weekly
# use the adm group by default, since this is the owning group
# of /var/log/.
su root adm
# keep 4 weeks worth of backlogs
rotate 4
# create new (empty) log files after rotating old ones
create
# use date as a suffix of the rotated file
#dateext
# uncomment this if you want your log files compressed
#compress
# packages drop log rotation information into this directory
include /etc/logrotate.d
# system-specific logs may also be configured here.
```



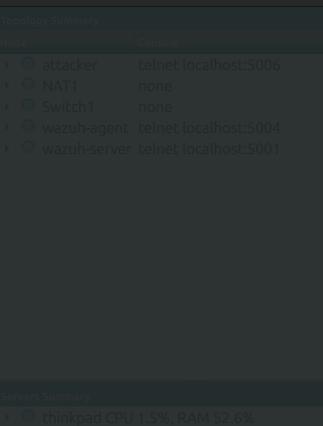
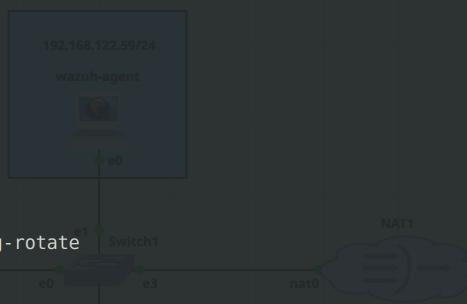
## • Program-specific configs

```
"=====
" Netrw Directory Listing
" /etc/logrotate.d
" Sorted by name
" Sort sequence: [ \ ]$, \<core\%(\.\d+\)\=\>, \.hs, \.c$, \.cpp$, \~\=*\$,*\.\.o$, \.obj$, \.info$, \.swp$, \.bak$, \~\$
" Quick Help: <F1>:help -:go up dir D:delete R:rename s:sort-by x:special
" =====
./
./
alternatives
apport
apt
bootlog
btmp
cloud-init
dpkg
rsyslog
ubuntu-pro-client
ufw
unattended-upgrades
wtmp
-
```



## • Example of a rsyslog logrotate config which I found on the wazuh-server

```
1 /var/log/syslog
2 /var/log/mail.log
3 /var/log/kern.log
4 /var/log/auth.log
5 /var/log/user.log
6 /var/log/cron.log
7 {
8     rotate 4
9     weekly
10    missingok
11    notifempty
12    compress
13    delaycompress
14    sharedscripts
15    postrotate
16        /usr/lib/rsyslog/rsyslog-rotate
17    endscript
18 }
```



## How disk space is managed in relation to logs

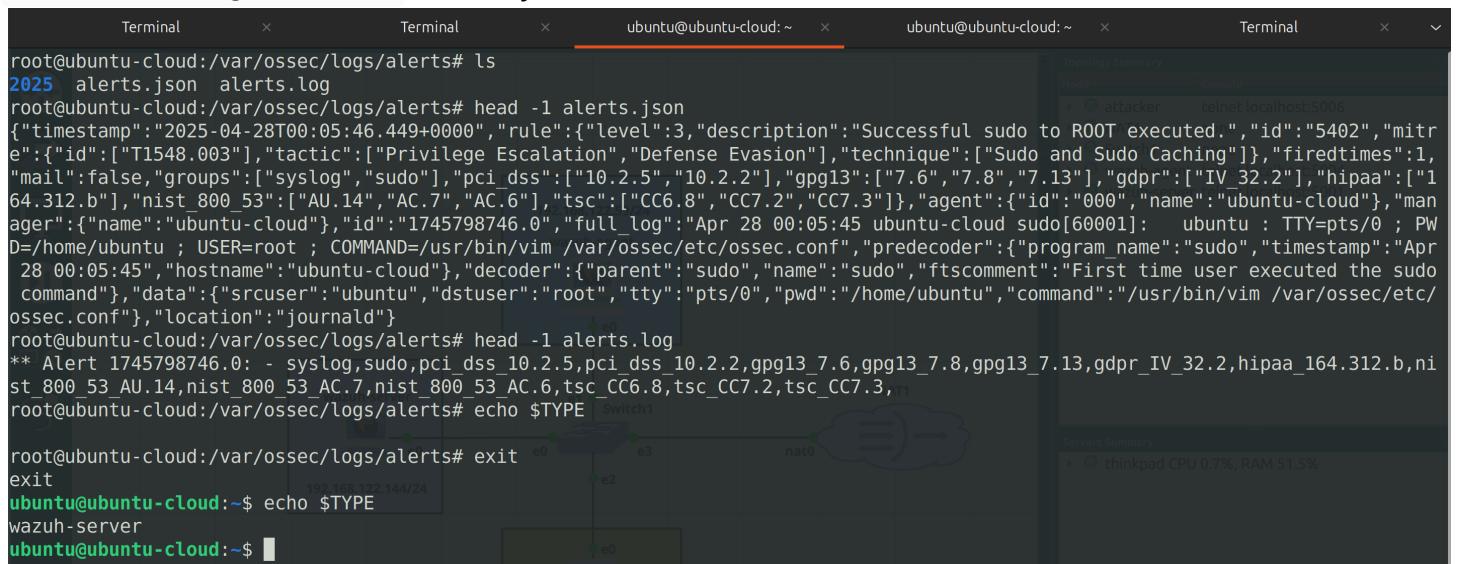
- **Space Consumption.** Logs can grow rapidly, depending on the amount of data generated by applications and services. Without proper management, logs can fill up disk space, leading to system performance issues and potential service interruptions.
- **Preventing Disk Full Conditions.** Using log rotation helps prevent disk full conditions by periodically cleaning up old log files. By specifying retention periods and ensuring that logs are compressed, organizations can effectively manage disk space usage.
- **Monitoring Disk Usage.** It's important to monitor disk usage statistics regularly to ensure that log rotation is effectively reducing space usage. Tools like `du`, `df`, and logging monitoring frameworks can help track how much space is consumed by logs.
- **Archiving and Offloading.** In some cases, organizations may need to archive old logs for future reference or compliance. This can involve transferring old log files to secondary storage or a centralized log management solution, such as using services like [ELK](#) stack, [Splunk](#), or [SageDB](#).

### 3.

#### Task description

Create a configuration to automatically delete Wazuh alert log files older than 90 days

- I have read that actually "By default, the Wazuh server retains logs and does not delete them automatically. However, you can choose when to manually or automatically delete these logs according to your legal and regulatory requirements." So that's why we actually may need to configure log retention!
- I explored that alert log files are stored on Wazuh Manager or `wazuh-server` in my case inside the `/var/ossec/logs/alerts/` directory



```

root@ubuntu-cloud:/var/ossec/logs/alerts# ls
2025 alerts.json alerts.log
root@ubuntu-cloud:/var/ossec/logs/alerts# head -1 alerts.json
{"timestamp": "2025-04-28T00:05:46.449+0000", "rule": {"level": 3, "description": "Successful sudo to ROOT executed.", "id": "5402", "mitre": {"id": "T1548.003"}, "tactic": ["Privilege Escalation", "Defense Evasion"], "technique": ["Sudo and Sudo Caching"]}, "firedtimes": 1, "mail": false, "groups": ["syslog", "sudo"], "pci_dss": ["10.2.5", "10.2.2"], "gpg13": ["7.6", "7.8", "7.13"], "gdpr": ["IV_32.2"], "hipaa": ["164.312.b"], "nist_800_53": ["AU.14", "AC.7", "AC.6"], "tsc": ["CC6.8", "CC7.2", "CC7.3"]}, "agent": {"id": "000", "name": "ubuntu-cloud"}, "manager": {"name": "ubuntu-cloud"}, "id": "1745798746.0", "full_log": "Apr 28 00:05:45 ubuntu-cloud sudo[60001]:    ubuntu : TTY=pts/0 ; PWD=/home/ubuntu ; USER=root ; COMMAND=/usr/bin/vim /var/ossec/etc/ossec.conf", "predecoder": {"program_name": "sudo", "timestamp": "Apr 28 00:05:45", "hostname": "ubuntu-cloud"}, "decoder": {"parent": "sudo", "name": "sudo", "ftscomment": "First time user executed the sudo command"}, "data": {"srcuser": "ubuntu", "dstuser": "root", "tty": "pts/0", "pwd": "/home/ubuntu", "command": "/usr/bin/vim /var/ossec/etc/ossec.conf"}, "location": "journald"}
root@ubuntu-cloud:/var/ossec/logs/alerts# head -1 alerts.log
** Alert 1745798746.0: - syslog,sudo,pci_dss_10.2.5,pci_dss_10.2.2,gpg13_7.6,gpg13_7.8,gpg13_7.13,gdpr_IV_32.2,hipaa_164.312.b,nist_800_53_AU.14,nist_800_53_AC.7,nist_800_53_AC.6,tsc_CC6.8,tsc_CC7.2,tsc_CC7.3,ri
root@ubuntu-cloud:/var/ossec/logs/alerts# echo $TYPE
root@ubuntu-cloud:/var/ossec/logs/alerts# exit
exit
ubuntu@ubuntu-cloud:~$ echo $TYPE
wazuh-server
ubuntu@ubuntu-cloud:~$ 

```

- Both `alerts.log` and `alerts.json` keep the same logs in a different format, while `2025` directory stores only compressed log files and their checksum that are signed daily

```

root@ubuntu-cloud:/var/ossec/logs/alerts/2025/Apr# ls -al
total 816
drwxr-x--- 2 wazuh wazuh 4096 Apr 28 21:17 .
drwxr-x--- 3 wazuh wazuh 4096 Apr 28 21:17 ..
-rw-r----- 1 wazuh wazuh 7820 Apr 28 00:00 ossec-alerts-27.json.gz
-rw-r----- 1 wazuh wazuh 499 Apr 28 00:00 ossec-alerts-27.json.sum
-rw-r----- 1 wazuh wazuh 5610 Apr 28 00:00 ossec-alerts-27.log.gz
-rw-r----- 1 wazuh wazuh 496 Apr 28 00:00 ossec-alerts-27.log.sum
-rw-r----- 2 wazuh wazuh 482305 Apr 28 21:32 ossec-alerts-28.json
-rw-r----- 2 wazuh wazuh 308128 Apr 28 21:32 ossec-alerts-28.log
root@ubuntu-cloud:/var/ossec/logs/alerts/2025/Apr#

```

- As we see both files have no information about Apr 27 logs

```

ubuntu@ubuntu-cloud:~$ sudo grep "Apr 27" /var/ossec/logs/alerts/alerts.json; sudo grep "Apr 27" /var/ossec/logs/alerts/alerts.log
ubuntu@ubuntu-cloud:~

```

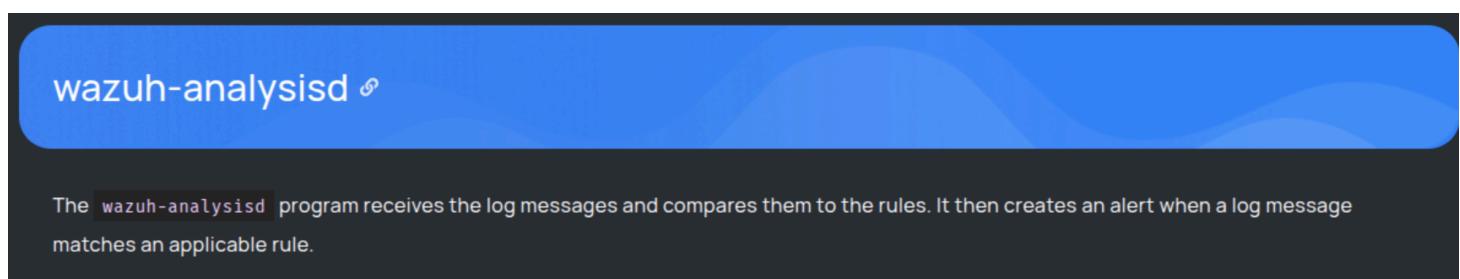
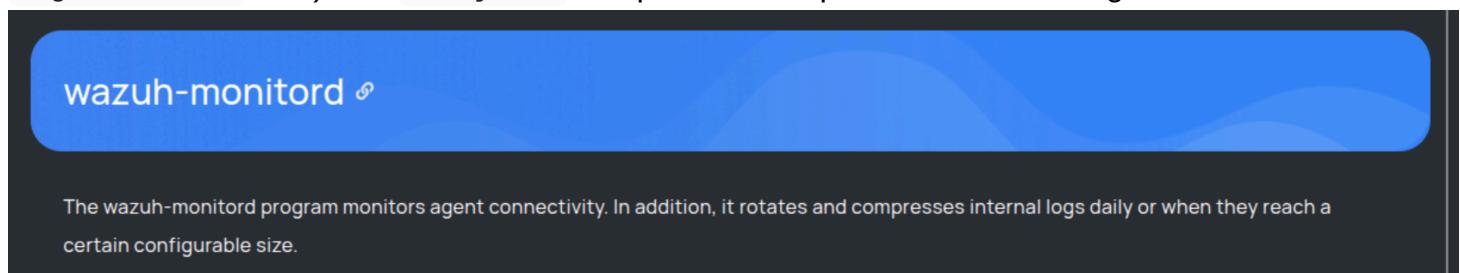
- While for Apr 28 (today) there are lots of log records

```

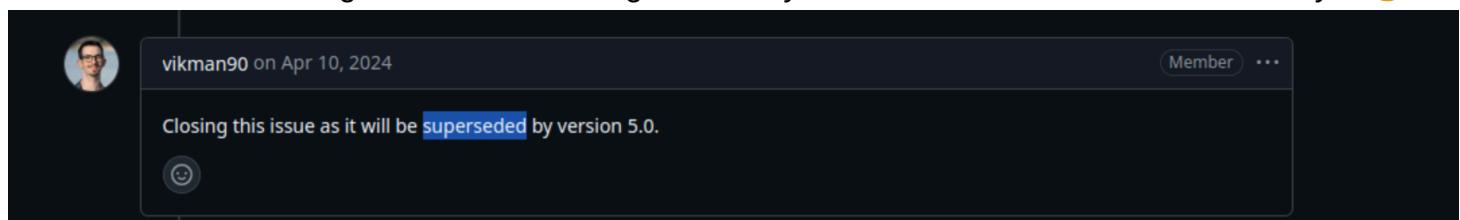
ubuntu@ubuntu-cloud:~$ sudo grep "Apr 28" /var/ossec/logs/alerts/alerts.json | head -1
{"timestamp": "2025-04-28T00:05:46.449+0000", "rule": {"level": 3, "description": "Successful sudo to ROOT executed.", "id": "5402", "mitre": {"id": "[T1548.003]", "tactic": ["Privilege Escalation", "Defense Evasion"], "technique": ["Sudo and Sudo Caching"]}, "firedtimes": 1, "mail": false, "groups": ["syslog", "sudo"], "pci_dss": ["10.2.5", "10.2.2"], "gpg13": ["7.6", "7.8", "7.13"], "gdpr": ["IV_32.2"], "hipaa": ["164.312.b"], "nist_800_53": ["AU.14", "AC.7", "AC.6"], "tsc": ["CC6.8", "CC7.2", "CC7.3"]}, "agent": {"id": "000", "name": "ubuntu-cloud"}, "manager": {"name": "ubuntu-cloud"}, "id": "1745798746.0", "full_log": "Apr 28 00:05:45 ubuntu-cloud sudo[60001]: ubuntu : TTY=pts/0 ; PWD=/home/ubuntu ; USER=root ; COMMAND=/usr/bin/vim /var/ossec/etc/ossec.conf", "predecoder": {"program_name": "sudo", "timestamp": "Apr 28 00:05:45", "hostname": "ubuntu-cloud"}, "decoder": {"parent": "sudo", "name": "sudo", "ftscomment": "First time user executed the sudo command"}, "data": {"srcuser": "ubuntu", "dstuser": "root", "tty": "pts/0", "pwd": "/home/ubuntu", "command": "/usr/bin/vim /var/ossec/etc/ossec.conf"}, "location": "journald"}
ubuntu@ubuntu-cloud:~$ sudo grep "Apr 28" /var/ossec/logs/alerts/alerts.log | head -1
2025 Apr 28 00:05:46 ubuntu-cloud->journald
ubuntu@ubuntu-cloud:~

```

- From all this I inferred that Wazuh Server already rotates logs daily! Then I found [proofs](#) of this inference. However, as far as I got this works only together `monitord` component which is responsible for "product" logs (`ossec.log`, `logs/wazuh`, not "event" (`logs/alerts`, `logs/archives`, etc.). But `analysisd` component is responsible for event logs.



- After I realized it, I tried to find some setting for Wazuh Server configuration to enable log retention for event logs. However, after a small research I found an official [issue](#) that states that log rotation mechanism for event logs will become configurable only in version 5.0 which is not released yet 😞



- Also, I haven't found a possibility to disable default log rotation policy. There were some [closed issues](#) only. Therefore, I cannot inject my own log rotation and retention mechanism such as **logrotate**
- Hence, the only option to make log retention I had is to write my own script making it a cron job.
- Below is the my script

```
#!/bin/bash

BASE_DIR="/var/ossec/logs/alerts"

# retention after 90 days
OLD_DATE=$(date -d '90 days ago' '+%Y-%b-%d')
OLD_YEAR=$(date -d "$OLD_DATE" '+%Y')
OLD_MONTH=$(date -d "$OLD_DATE" '+%b')
OLD_DAY=$(date -d "$OLD_DATE" '+%d')

find "$BASE_DIR/$OLD_YEAR/$OLD_MONTH" -type f -name "ossec-alerts-$OLD_DAY.*" -exec rm -f {} \;
```

- And here is the configuration of a cronjob that runs my script 3 times a day (at 6am, 12pm, 6pm) every day.

```
0 6,12,18 * * * /path/to/your/script/clean_old_logs.sh
```

- So, I created my script on the wazuh-server

The screenshot shows a terminal session on an Ubuntu Cloud instance. The user has opened four terminal windows. The active window displays the contents of a file named `wazuh_log_retention.sh`. The script is a shell script that defines a variable `BASE_DIR` and then uses the `find` command to locate old log files in the directory `/var/ossec/logs/alerts` and delete them. The script is then made executable with `chmod +x`, and its path is checked with `ls -l`.

```
ubuntu@ubuntu-cloud:~$ vim wazuh_log_retention.sh
ubuntu@ubuntu-cloud:~$ cat wazuh_log_retention.sh
#!/bin/bash

BASE_DIR="/var/ossec/logs/alerts"

# retention after 90 days
OLD_DATE=$(date -d '90 days ago' '+%Y-%b-%d')
OLD_YEAR=$(date -d "$OLD_DATE" '+%Y')
OLD_MONTH=$(date -d "$OLD_DATE" '+%b')
OLD_DAY=$(date -d "$OLD_DATE" '+%d')

find "$BASE_DIR/$OLD_YEAR/$OLD_MONTH" -type f -name "ossec-alerts-$OLD_DAY.*" -exec rm -f {} \;

ubuntu@ubuntu-cloud:~$ chmod +x wazuh_log_retention.sh
ubuntu@ubuntu-cloud:~$ ls -l wazuh_log_retention.sh
-rwxrwxr-x 1 ubuntu ubuntu 331 Apr 28 23:55 wazuh_log_retention.sh
ubuntu@ubuntu-cloud:~$
```

- And created a cronjob

```

ubuntu@ubuntu-cloud:~$ crontab -e
no crontab for ubuntu - using an empty one
Select an editor. To change later, run select-editor again.
 1. /bin/nano   <---- easiest
 2. /usr/bin/vim.basic
 3. /usr/bin/vim.tiny
 4. /bin/ed

Choose 1-4 [1]: 2
crontab: installing new crontab
ubuntu@ubuntu-cloud:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
0 6,12,18 * * * /path/to/your/script/clean_old_logs.sh
ubuntu@ubuntu-cloud:~$
```

## Bonus

### 1.

#### Task description

What are indices, and how do they differ from log files?

## Definition

- An **index** in Elasticsearch or similar log database (such as Wazuh Indexer) is a collection of documents that share similar characteristics. It is akin to a database in traditional relational database management systems. Each index is essentially a **data structure optimized for fast search and retrieval operations**. Indices hold documents in a format that Elasticsearch can efficiently process. Each document represents a discrete piece of data (like a JSON object) and can contain various fields with associated values.

## Comparison to log files

- **Purpose:**

- **Log Files:** Serve as raw data sources where events and logs are generated. They are the initial form of data storage before any processing.

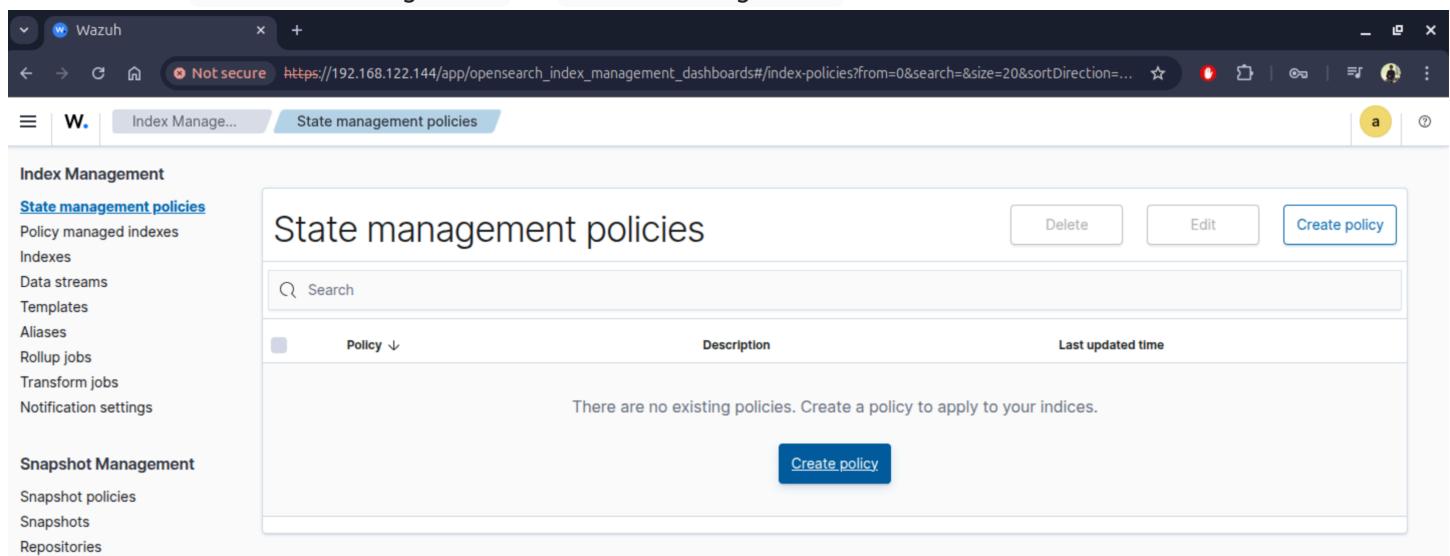
- **Indices:** Serve as organized collections of processed data that enable fast querying and analysis. They allow for efficient searching and aggregations based on structured data.
- **Structure:**
  - **Log Files:** Typically unstructured or semi-structured text files, which can be difficult to query without processing.
  - **Indices:** Structured data storage with defined mappings and fields, optimized for rapid searches and analyses based on those fields.
- **Lifecycle:**
  - **Log Files:** May be stored on disk in a particular location or directory for a specified duration, governed by log rotation policies.
  - **Indices:** Managed and retained according to index lifecycle policies in a database, which can include retention, aliasing, and deletion strategies.
- **Querying:**
  - **Log Files:** Queries against raw log files require additional tools or scripts to search through plain text data.
  - **Indices:** Queries against indices can leverage Elasticsearch's powerful search capabilities, allowing for precise queries and visualizations.

## 2.

### Task description

Create an index retention policy to delete Wazuh alert indices after 90 days.

- As always I found an [instruction](#) on the Wazuh official website
- I went to the Indexer Management -> Index Management



The screenshot shows the Wazuh web interface with the URL [https://192.168.122.144/app/opensearch\\_index\\_management\\_dashboards#/index-policies?from=0&search=&size=20&sortDirection=...](https://192.168.122.144/app/opensearch_index_management_dashboards#/index-policies?from=0&search=&size=20&sortDirection=...). The left sidebar has sections for Index Management (selected), State management policies, Policy managed indexes, Indexes, Data streams, Templates, Aliases, Rollup jobs, Transform jobs, and Notification settings. The main content area is titled "State management policies" and contains a table with one row: "Policy ↓", "Description", and "Last updated time". Below the table, a message says "There are no existing policies. Create a policy to apply to your indices." A "Create policy" button is at the bottom right.

- I was suggested to select a method to create a new policy:
  - Visual editor in Web UI
  - Or in JSON editor

- Using a visual editor I created a new `test-policy` policy and configured two states for `wazuh-alerts-*` indices:
  - `initial`: logs are initially in this state
  - `delete`: after 90 days logs are transited to this state
- After a transition to `delete` state logs are immediately deleted
- The result is below

The screenshot shows the Wazuh OpenSearch Index Management Dashboard with the URL `https://192.168.122.144/app/opensearch_index_management_dashboards#/policy-details?id=test-policy`. The page displays the `test-policy` settings, ISM Templates, and States.

### Policy settings

Policy name	Error notification	Primary term	Last updated
test-policy	-	1	4/28/2025, 11:22:45 PM

**Policy description:** A policy to clear logs older than 90 days

### ISM Templates (1)

Index patterns	Priority
wazuh-alerts-*	1

Rows per page: 10 < 1 >

### States (2)

You can think of policies as state machines. "Actions" are the operations ISM performs when an index is in a certain state. "Transitions" define when to move from one state to another. [Learn more](#)

Initial State: `initial`

Actions: `delete_alerts`

Transitions:

- `initial` → `delete_alerts`

- Then I simply selected to modify the policy this time using the JSON editor

The screenshot shows the Wazuh OpenSearch Index Management Dashboard. In the center, a modal dialog titled "Configuration method" is displayed, asking how to modify the policy (Visual editor or JSON editor). The "JSON editor" option is selected and highlighted with a blue circle. The background shows the "test-policy" configuration page with sections for "Policy settings", "ISM Templates (1)", and "States (2)".

- The JSON editor was appeared and I could see the configuration of my log retention policy in a JSON format

The screenshot shows the Wazuh OpenSearch Index Management Dashboard with the "Edit policy" tab selected for the "test-policy". The "Define policy" section displays the JSON configuration of the log retention policy. The JSON code is as follows:

```

1  {
2    "policy": {
3      "policy_id": "test-policy",
4      "description": "A policy to clear logs older than 90 days",
5      "last_updated_time": 1745871765015,
6      "schema_version": 21,
7      "error_notification": null,
8      "default_state": "initial",
9      "states": [
10        {
11          "name": "initial",
12          "actions": [],
13          "transitions": [
14            {
15              "state_name": "delete_alerts",
16              "conditions": [
17                "min_index_age": "90d"
18              ]
19            }
20          ]
21        },
22        {
23          "name": "delete_alerts",
24          "actions": []
25        }
26      ]
27    }
28  }

```

- So, below is the configuration of my log retention policy

```
{
  "policy": {
    "policy_id": "test-policy",
    "description": "A policy to clear logs older than 90 days",
    "last_updated_time": 1745871765015,
    "schema_version": 21,
    "error_notification": null,
    "default_state": "initial",
    "states": [
      {
        "name": "initial",
        "actions": [],
        "transitions": [
          {
            "state_name": "delete_alerts",
            "conditions": {
              "min_index_age": "90d"
            }
          }
        ]
      },
      {
        "name": "delete_alerts",
        "actions": [
          {
            "retry": {
              "count": 3,
              "backoff": "exponential",
              "delay": "1m"
            },
            "delete": {}
          }
        ],
        "transitions": []
      }
    ],
    "ism_template": [
      {
        "index_patterns": [
          "wazuh-alerts-*"
        ],
        "priority": 1,
        "last_updated_time": 1745871200545
      }
    ]
  }
}
```

- Then I applied this policy to the existing wazuh-alerts-\* indices manually

Indexes (14)

Index	Health	Managed by p...	Status	Total size	Size of primari...	Total documen...
wazuh-statistics-2025.18w	Green	No	Open	225.2kb	225.2kb	132
wazuh-statistics-2025.17w	Green	No	Open	146.7kb	146.7kb	64
wazuh-states-vulnerabilities-ubuntu-cloud	Green	No	Open	132.3kb	132.3kb	22
wazuh-monitoring-2025.18w	Green	No	Open	108.6kb	108.6kb	23
wazuh-monitoring-2025.17w	Green	No	Open	146.9kb	146.9kb	8
wazuh-alerts-4.x-2025.04.28	Green	No	Open	954.5kb	954.5kb	581
wazuh-alerts-4.x-2025.04.27	Green	No	Open	551.6kb	551.6kb	141
.ql-datasources	Green	No	Open	208b	208b	0
.plugins-ml-config	Green	No	Open	3.9kb	3.9kb	1
.opensearch-observability	Green	No	Open	208b	208b	0
.opendistro_security	Green	No	Open	78.4kb	78.4kb	10
.opendistro-reports-Instances	Green	No	Open	208b	208b	0
.opendistro-reports-Definitions	Green	No	Open	208b	208b	0

Indexes (14)

Apply policy

Choose the policy you want to use for the selected indices. A copy of the policy will be created and applied to the indices.

Policy ID

test-policy |

Preview

```
{
  "policy": {
    "policy_id": "test-policy",
    "description": "A policy to clear logs older than 90 days",
    "last_updated_time": 1745871765015,
    "schema_version": 21,
    "error_notification": null,
    "default_state": "initial",
    "states": [
      {
        "name": "initial"
      }
    ]
  }
}
```

Cancel      Apply

- Finally, I could see that my policy was applied to the wazuh-alerts indices

The screenshot shows the Wazuh OpenSearch interface. At the top, there's a navigation bar with tabs for 'Index Manage...', 'Indexes', and the selected index 'wazuh-alerts-4.x-2025.04.28'. On the right side of the header, there are several icons for search, refresh, and other functions.

The main content area is titled 'wazuh-alerts-4.x-2025.04.28'. Below this, there's an 'Overview' section with various metrics:

Index name	Health	Status
wazuh-alerts-4.x-2025.04.28	Green	Open
Creation date	Total size	Size of primaries
4/28/2025, 3:05:55 AM	954.5kb	954.5kb
Total documents	Deleted documents	Primaries
581	0	3
Replicas	Index blocks	Managed by policy
0	-	test-policy

Below the overview, there are three tabs: 'Settings', 'Mappings', and 'Alias'. The 'Alias' tab is currently selected. It contains a section for 'Index alias' with the following text:

**Index alias – optional**  
Allow this index to be referenced by existing aliases or specify a new alias.

A dropdown menu labeled 'Select aliases or specify new aliases.' is shown.