

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

## LAB4 SIEM

### Description

This lab is designed to introduce students to security solutions, specifically a SIEM. In this lab, students can use any SIEM of choice; regardless, a solid recommendation is to use the open source security platform Wazuh as this provides a fleet of capabilities at no cost.

In this lab, students will interact with additional tools such as virustotal, YARA, osquery, SOAR and also gain experience with SIEM log analysis, vulnerability detection and more.

## Part A

---

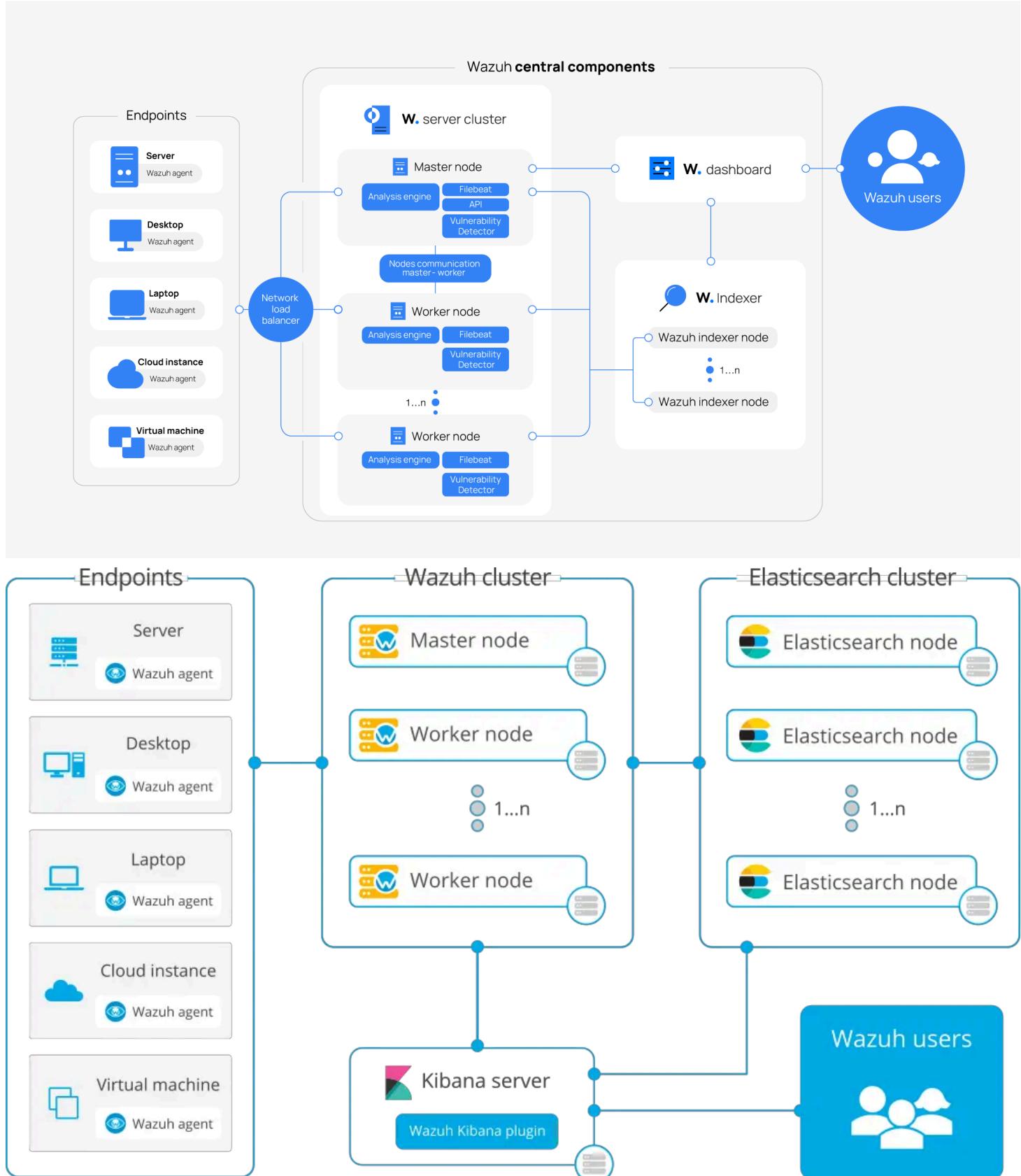
### Task 1 - Introduction

a.

#### Task description

- Give a brief explanation of the architecture of your SIEM solution

- Here is the pictures



- Below is the description:

- **Wazuh Agents:** These lightweight agents are installed on the endpoints that need to be monitored. They collect security event data, such as logs, and system information, perform file integrity checks, and monitor for vulnerabilities. The agents then send this data to the Wazuh manager.
- **Wazuh Manager:** This central component aggregates and analyzes the data received from the agents. It processes alerts, applies rules for threat detection, and generates security notifications.

The Wazuh manager also communicates with the Wazuh API and the Wazuh dashboard.

- **Wazuh Indexer:** Wazuh utilizes indexers for storing and indexing the log data collected from agents. It enables powerful search capabilities and facilitates the real-time querying of security logs and events.
  - **Wazuh Dashboard:** This is a visualization tool that is used to create dashboards and reports based on the data stored in Wazuh Indexers. It provides a user-friendly interface for visualizing security data, allowing users to analyze and correlate events effectively.
  - **Wazuh API:** The API allows for integration with other tools and automation workflows. It provides programmatic access to the Wazuh manager's functionality, enabling users to query alerts, manage agents, and retrieve logs.
- You may also notice that [ELK](#) stack can be used instead of Wazuh Indexers and Wazuh Dashboard.

b.

### Task description

- b. Provide 3 advantages of open source solutions and how do these vendors actually make money?

- **Advantages:**

1. **Cost Savings:**

- Open source software is **typically** (but not always!) free to use, which can significantly reduce the costs associated with software licensing. Organizations can allocate their budgets towards other critical areas, such as infrastructure, training, or personnel.

2. **Flexibility and Customization:**

- Open source solutions allow users to access and modify (not always!) the source code, enabling organizations to customize the software to fit their specific needs. This flexibility allows for the development of unique features, integrations, and functionality that may not be available in proprietary software.

3. **Community Support and Innovation:**

- Open source projects often have vibrant communities that contribute to their development, maintenance, and support. This collaborative environment can lead to rapid innovation, as many developers contribute ideas, report issues, and create enhancements. Additionally, community forums and user groups provide valuable resources for troubleshooting and best practices.

- **How do vendors make money?**

- ! Firstly, [Open Source](#) does not immediately mean [free software](#) (according to Richard Stallman), so companies may make code of their product open source while keeping it proprietary. Therefore companies can sell their software that is open-source directly to customers.
- **Creation for their own sake.** Some big corporations may need some software that they develop and make open-source after to make the technology developing further. In such a case this

software is not their primary goal to earn money but rather a tool that they make available to the community.

- **Consulting and Professional Services.** Open source vendors may offer consulting services, training, and implementation assistance to help organizations effectively deploy and utilize their software.
- **Hosted Solutions.** Some open source vendors provide cloud-hosted versions of their software as a service (SaaS). Customers pay for the convenience of using the software without the need for local installations or maintenance.
- **Donations:** Open source vendors can generate revenue through donations by allowing individuals and organizations to contribute financially to the projects they use. This enables communities to support ongoing development, maintenance, and feature enhancement.

## Task 2 - Setup infrastructure

a.

### Task description

- Configure a SIEM solution with 3(or more) unique devices. e.g Windows, Linux and a Network device. Can you view log data from each connected device? If yes show this.

## Wazuh server

- Firstly, I setup Wazuh server-side: manager, indexer, dashboard on the cloud machine using Docker
- Below is a configuration of my cloud VM wazuh-server-ubuntu on [Yandex Cloud](#)

The screenshot shows the Yandex Cloud interface for managing virtual machines. At the top, there's a navigation bar with 'cloud-iskanred' and 'iu-sd-course' dropdowns, a search bar, and a 'Create virtual machine' button. Below the header, the 'Virtual machines' section is displayed. A search bar at the top of this section contains the text 'wazuh-server-ubuntu'. There are several filter buttons: 'All statuses', 'All platforms', 'All availability zones', and a 'Clear' button. A table lists the single virtual machine: 'wazuh-server-ubuntu' (Status: Running, OS: Intel Ice Lake, Platform: vCPU, RAM: 8 GB, Disk size: 50 GB, Availability zone: ru-central1-b, Internal IPv4: 10.129.0.33, Public IPv4: 89.169.173.179, Date created: 03/03/2025, at 2:...).

- IP = 89.169.173.179 , local user is nafikov , and hostname is wazuh-server-ubuntu
- I installed Docker there and cloned the official wazuh-docker Git repo:  
<https://github.com/wazuh/wazuh-docker/tree/main/single-node>

```
nafikov@wazuh-server-ubuntu:~/wazuh-docker/single-node/config$ ls certs.yml wazuh_cluster wazuh_dashboard wazuh_indexer wazuh_indexer_ssl_certs
nafikov@wazuh-server-ubuntu:~/wazuh-docker/single-node/config$ git status
warning: could not open directory 'single-node/config/wazuh_indexer_ssl_certs/': Permission denied
HEAD detached at v4.11.0
nothing to commit, working tree clean
nafikov@wazuh-server-ubuntu:~/wazuh-docker/single-node/config$
```

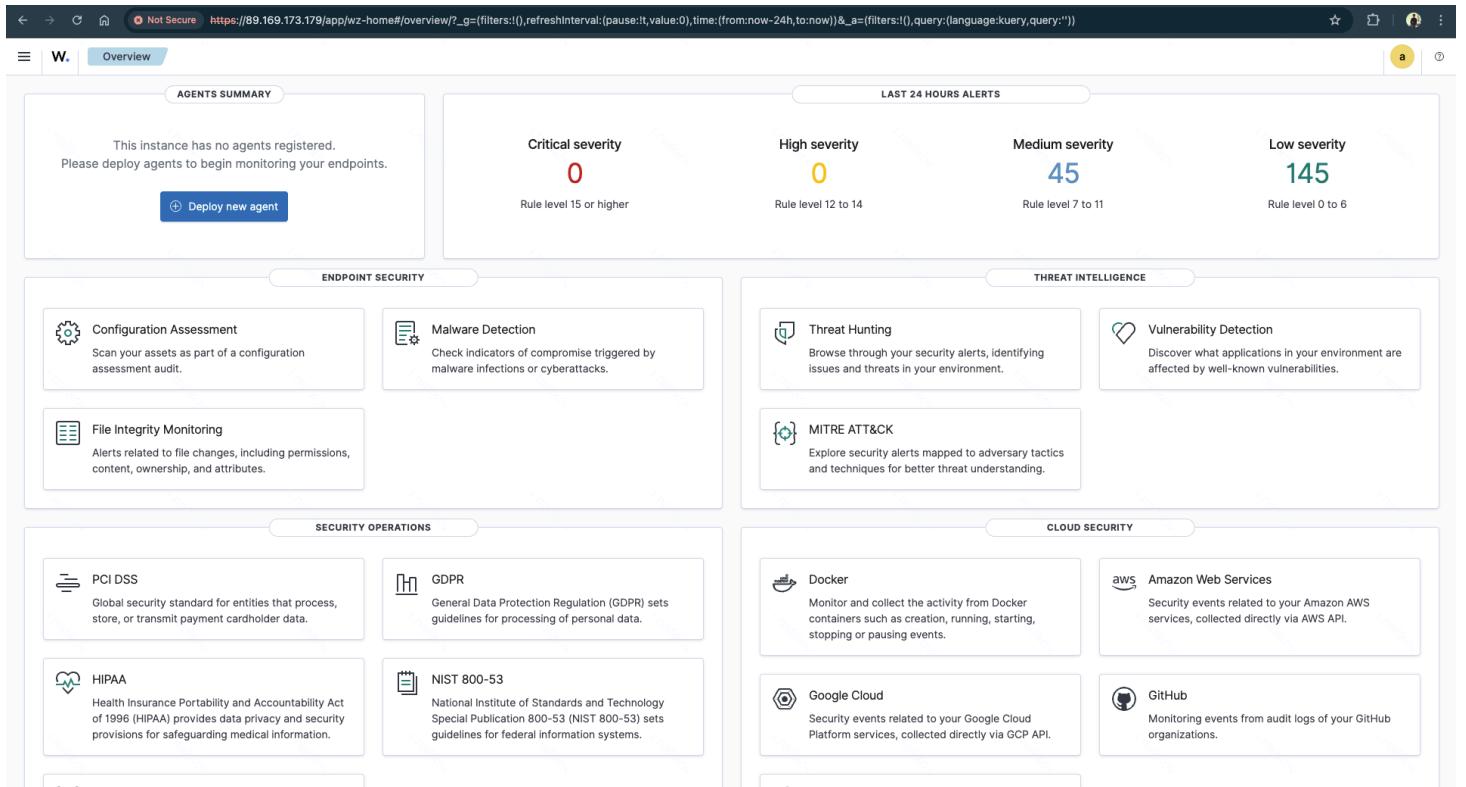
- Following the instruction in single-node/README.md I generated TLS/SSL certificates and finally ran the compose file

```

-- nafikov@wazuh-server-ubuntu: ~/wazuh-docker/single-node/config$ docker compose ls
NAME      STATUS    CONFIG FILES
single-node  running(3)   /home/nafikov/wazuh-docker/single-node/docker-compose.yml
nafikov@wazuh-server-ubuntu:~/wazuh-docker/single-node/config$ docker ps
CONTAINER ID IMAGE          COMMAND           CREATED          STATUS          PORTS
cb9268748949  wazuh/wazuh-dashboard:4.11.0  "/entrypoint.sh"  12 minutes ago  Up 12 minutes  443/tcp, 0.0.0.0:443->5601/tcp, [::]:443->5601/tcp
84ef0dd0b071  wazuh/wazuh-manager:4.11.0   "/init"        12 minutes ago  Up 12 minutes  0.0.0.0:1514->1514-1515/tcp, [::]:1514-1515->1514-1515/tcp, 0.
a22d2b05f5a1  wazuh/wazuh-indexer:4.11.0   "/entrypoint.sh open..."  12 minutes ago  Up 12 minutes  0.0.0.0:9200->9200/tcp, [::]:9200->9200/tcp
nafikov@wazuh-server-ubuntu:~/wazuh-docker/single-node/config$ 

```

- Finally, I was able to access the dashboard:



## Wazuh agents

### Windows

- First, I installed an agent on my **Windows machine** using the PowerShell script:

```

Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.11.0-1.msi -OutFile $env:tmp\wazuh-agent
msiexec.exe /i $env:tmp\wazuh-agent /q WAZUH_MANAGER='89.169.173.179'
WAZUH_AGENT_GROUP='default' WAZUH_AGENT_NAME='windows-agent'

```

```

PS C:\Users\liskanned> Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.11.0-1.msi -OutFile $env:tmp\wazuh-agent; msiexec.exe /i $env:tmp\wazuh-agent /q WAZUH_MANAGER='89.169.173.179' WAZUH_AGENT_GROUP='default' WAZUH_AGENT_NAME='windows-agent'
PS C:\Users\liskanned> NET START WazuhSvc
The Wazuh service is starting.
The Wazuh service was started successfully.
PS C:\Users\liskanned> 

```

- Finally I could see my first endpoint in Wazuh

The screenshot shows the Wazuh UI interface. At the top, there are three donut charts: 'AGENTS BY STATUS' (Active 1), 'TOP 5 OS' (windows 1), and 'TOP 5 GROUPS' (default 1). Below these are sections for 'Agents (1)' and 'File Integrity Monitoring'.

**Agents (1)**

- Show only outdated
- WQL

ID ↑	Name	IP address	Group(s)	Operating system	Cluster node	Version	Status	Actions
001	windows-agent	172.20.10.7	default	Microsoft Windows 10 Enterprise LTSC 2019 10.0.17763.6659	node01	v4.11.0	active ⓘ	...

Rows per page: 10 < 1 >

- We can see its state from the perspective of Wazuh

The screenshot shows the Wazuh UI interface with several panels:

- Threat Hunting**, **File Integrity Monitoring**, **Configuration Assessment**, **MITRE ATT&CK**, **Vulnerability Detection**, **More...**
- Events count evolution**: A line chart showing event counts over time.
- MITRE ATT&CK**: A dashboard showing Top Tactics (Persistence, Privilege Escalation, Defense Evasion, Initial Access) with counts (21, 21, 20, 19).
- Compliance**: A donut chart showing PCI DSS compliance levels (2.2, 2.2.5, 4.1, 10.6.1, 7.1).
- Vulnerability Detection**: A summary of findings:
  - 1 Critical
  - 80 High
  - 87 Medium
  - 24 Low
- Top 5 Packages** table:
 

Package	Count ↓
Foxit Reader	120
Wireshark 3.6.1 64-bit	35
Python 3.10.2 (64-bit)	18
Steam	8
Notepad++ (32-bit x86)	4
- SCA: Lastest scans**: CIS Microsoft Windows 10 Enterprise Benchmark v1.12.0 (cis\_win10\_enterprise):
 

Policy	End scan	Passed	Failed	Not applic...	Score
CIS Microsoft Windows 10 Enterprise Benchmark v1.12.0	Mar 6, 2025 @ 09:27:41.000	128	262	4	32%

## Ubuntu

- Now let's integrate Wazuh agent to my **virtual Ubuntu Cloud machine**:

```
wget https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-
agent_4.11.0-1_amd64.deb && \
sudo WAZUH_MANAGER='89.169.173.179' WAZUH_AGENT_GROUP='default'
WAZUH_AGENT_NAME='ubuntu-agent' dpkg -i ./wazuh-agent_4.11.0-1_amd64.deb
```

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

```

iskanred@lenovo:~                                         iskanred@lenovo:~                                         ubuntu-agent
ubuntu@ubuntu-cloud:~$ sudo wget https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.11.0-1_amd64.deb
wget https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.11.0-1_amd64.deb && sudo WAZUH_MANAGER='89.169.173.179' WAZUH_AGENT_GROUP='default' WAZUH_AGENT_NAME='ubuntu-agent' dpkg -i ./wazuh-agent_4.11.0-1_amd64.deb
--2025-03-06 15:26:43-  https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.11.0-1_amd64.deb
Resolving packages.wazuh.com (packages.wazuh.com)... 3.164.68.88, 3.164.68.26, 3.164.68.41, ...
Connecting to packages.wazuh.com (packages.wazuh.com)|3.164.68.88|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11078824 (11M) [application/vnd.debian.binary-package]
Saving to: 'wazuh-agent_4.11.0-1_amd64.deb'

wazuh-agent_4.11.0- 100%[=====] 10.57M 117KB/s   in 92s

2025-03-06 15:28:16 (118 KB/s) - 'wazuh-agent_4.11.0-1_amd64.deb' saved [11078824/11078824]

Selecting previously unselected package wazuh-agent.
(Reading database ... 76019 files and directories currently installed.)
Preparing to unpack .../wazuh-agent_4.11.0-1_amd64.deb ...
Unpacking wazuh-agent (4.11.0-1) ...
Setting up wazuh-agent (4.11.0-1) ...
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:~$ sudo systemctl status wazuh-agent.service
● wazuh-agent.service - Wazuh agent
   Loaded: loaded (/usr/lib/systemd/system/wazuh-agent.service; enabled; pres
   Active: active (running) since Thu 2025-03-06 15:30:31 UTC; 21s ago
     Invocation: 361d8092740841f0ae29364ad4714d66
      Process: 2914 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (co
     Tasks: 22 (limit: 1058)
    Memory: 131.3M (peak: 131.9M)
       CPU: 3.918s
      CGroup: /system.slice/wazuh-agent.service
              └─3079 /bin/sh active-response/bin/restart.sh agent

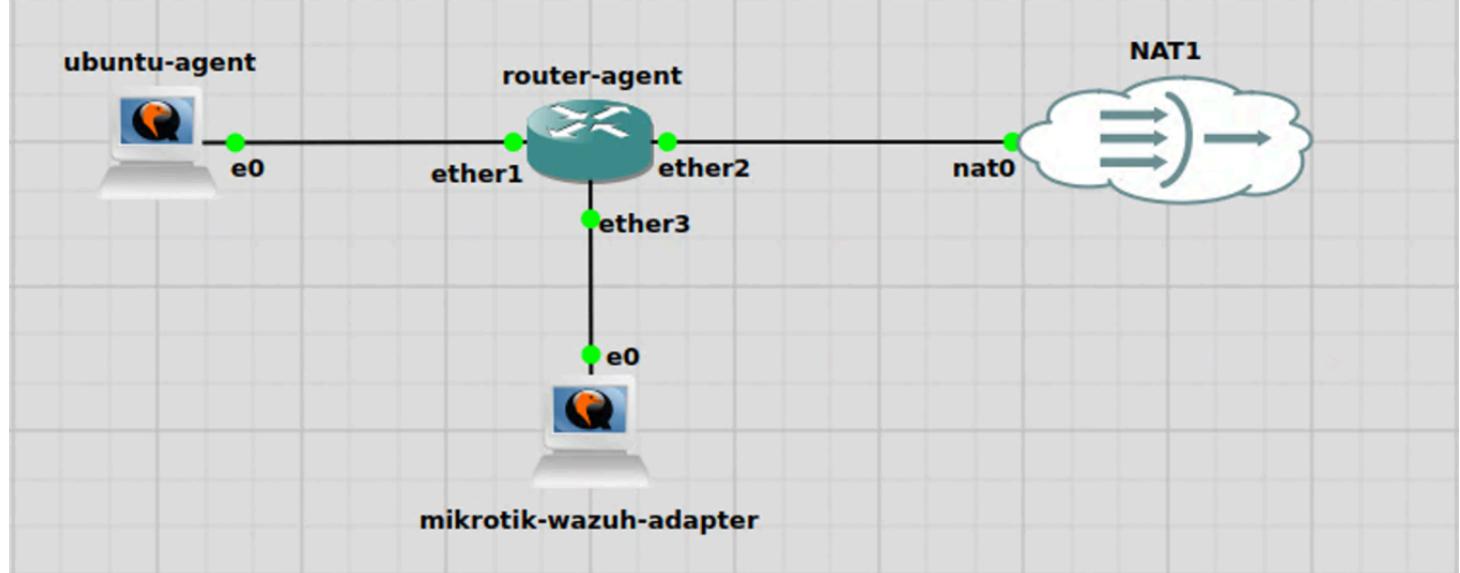
ubuntu@ubuntu-cloud:~$
```

The network diagram illustrates a topology where a central router (labeled 'router-agent') is connected to two hosts: 'ubuntu-agent' and 'mikrotik-wazuh-adapter'. The 'ubuntu-agent' host has an interface 'e0' connected to the router's 'ether1'. The 'mikrotik-wazuh-adapter' host also has an interface 'e0' connected to the router's 'ether1'. The router itself has three interfaces: 'ether2' (connected to 'nat0'), 'ether3' (connected to the 'mikrotik-wazuh-adapter' host), and 'nat0' (connected to a cloud labeled 'NAT1'). The 'nat0' interface is shown with three outgoing arrows pointing away from the cloud.

Agents (1)							
ubuntu							
ID	Name	IP address	Group(s)	Operating system	Cluster node	Version	Status
002	ubuntu-agent	192.168.0.2	default	Ubuntu 24.10	node01	v4.11.0	● active ⓘ
Rows per page: 10							

## Mikrotik Router

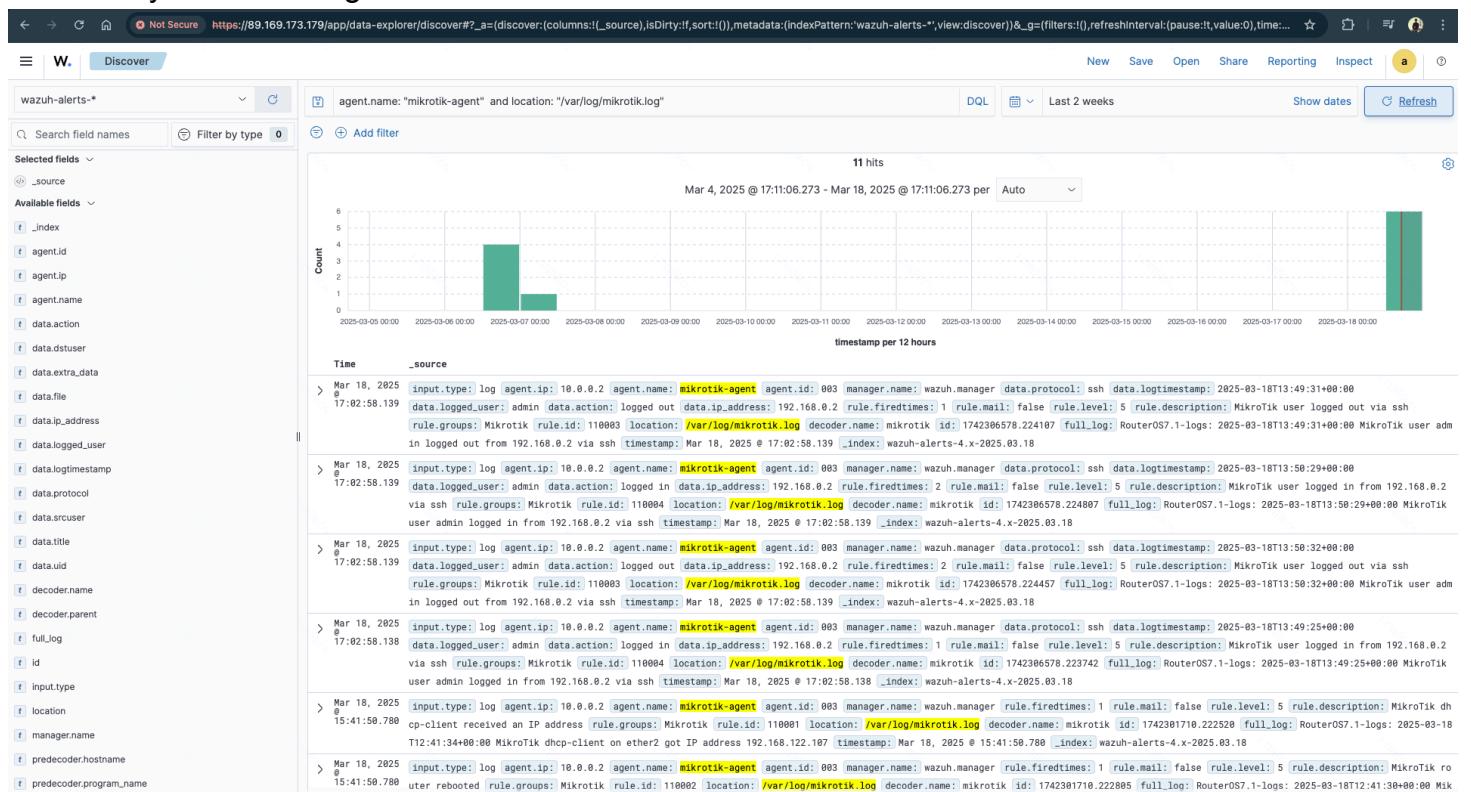
- Finally, let's connect my **virtual Mikrotik Router**
- Using [this](#) official article I configured transferring logs from the router to the Wazuh server via another Ubuntu agent ( `mikrotik-wazuh-adapter` ).
- By the way I found some mistake in this article and created an [issue](#) on GitHub for wazuh-documentation
- Therefore, I got the following network structure:



- We have `ubuntu-agent` , `router-agent` , and `mikrotik-wazuh-adapter` .
- The following log events will be sent from the router:
  - MikroTik dhcp-client received an IP address

- MikroTik router rebooted
- MikroTik user logged out via {PROTOCOL}
- MikroTik user logged in from {IP\_ADDRESS} via {PROTOCOL}

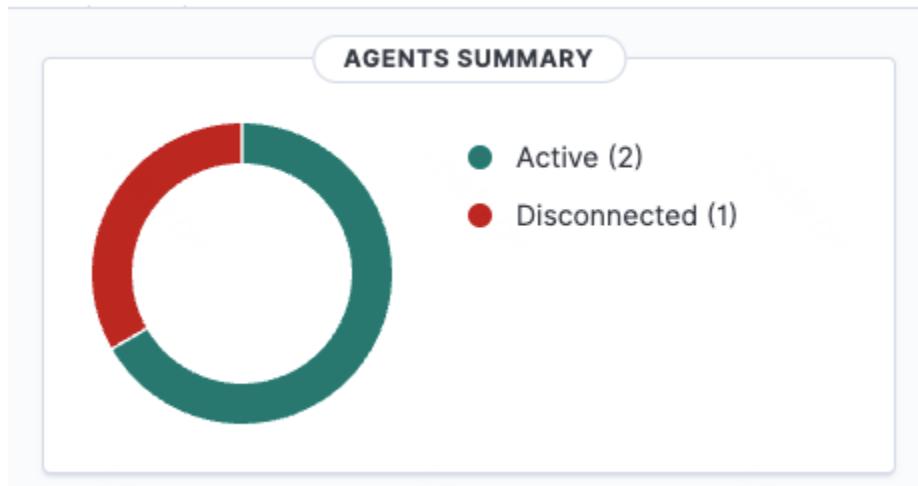
- Let's finally check the logs in Wazuh



- We see that sending logs actually works

## ⚠ Disclaimer

- Unfortunately, I have **two systems** (Windows & Ubuntu) **on the same machine**, so I need to disable Windows system (therefore making this endpoint inactive). I cannot use virtualisation for 3 devices since I have only 4Gb of memory on my device 😢
- So, I decided to keep only these two systems and connect Windows only when necessary.

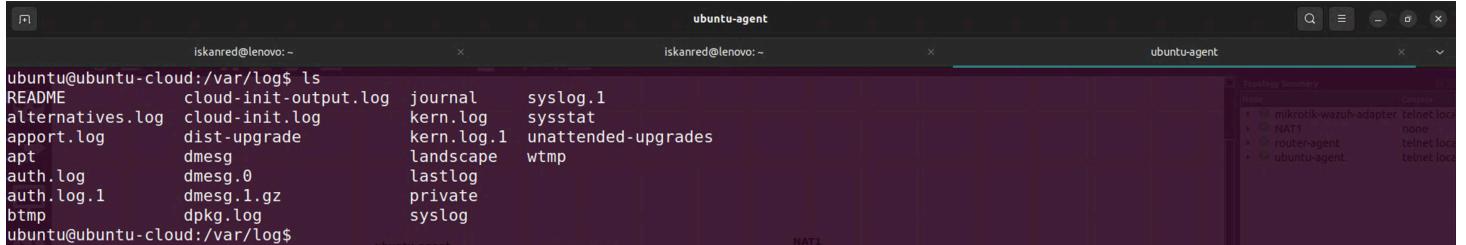


b.

## 🔗 Task description

b. Why specifically are you able to view these logs i.e select two visible logs, explain these logs, and explain why and how you are able to view it on the SIEM.

- I am able to view these logs because they were sent to the Wazuh Manager Server using `wazuh-agent` software. These logs are scrapped using `rsyslog`. For example there are some log files on `ubunut-agent`:



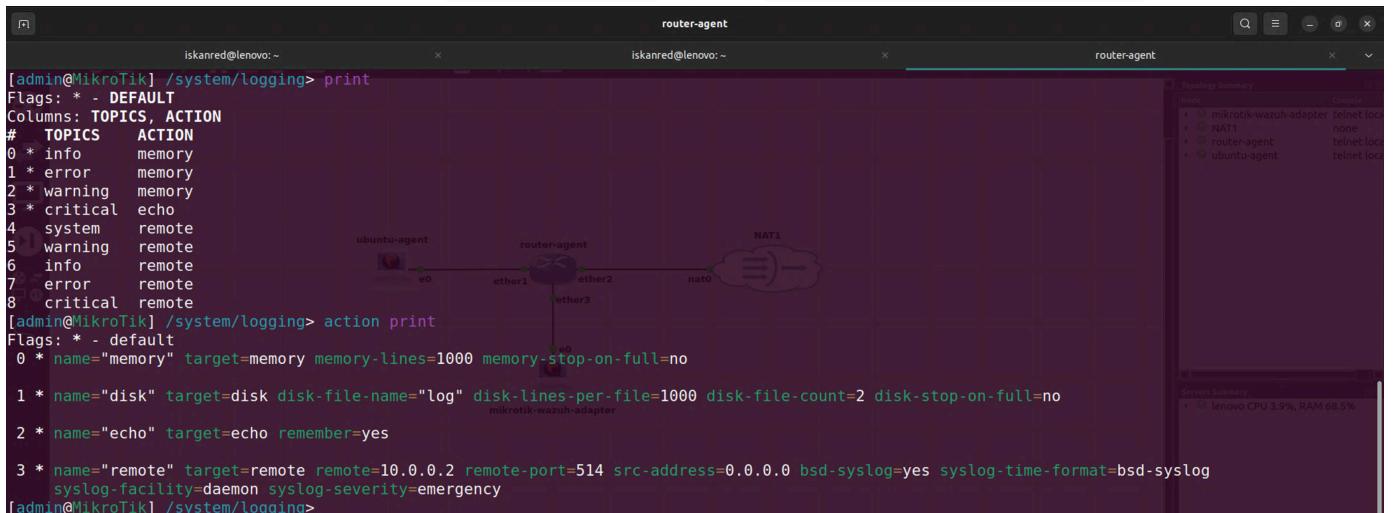
The screenshot shows a terminal window titled "ubuntu-agent" with the command `ls /var/log` running. The output lists several log files including `alternatives.log`, `apt.log`, `auth.log`, `auth.log.1`, `btmp`, `cloud-init-output.log`, `dmesg`, `dmesg.0`, `dmesg.1.gz`, `dpkg.log`, `journal`, `kern.log`, `lastlog`, `private`, `syslog`, and `syslog.1`. To the right of the terminal, a "Topology Summary" pane shows a network diagram with nodes: "mikrotik-wazuh-adapter", "NAT1", "router-agent", and "ubuntu-agent". The "ubuntu-agent" node is highlighted in blue. A "Console" pane on the far right shows the same network diagram with "telnet loc" listed under each node.

- Let's check two different logs and try to explain them

### MikroTik user logged in from {IP\_ADDRESS} via {PROTOCOL}

- Explanation:** This log tells that someone logged in to the router's system via some protocol from some IP address.
- Why and how I am able to view it:**

- Mikrotik logs are sent from the `router-agent` to the `mikrotik-wazuh-adapter`.



The screenshot shows a terminal window titled "router-agent" with the command `[admin@MikroTik] /system/logging> print` running. The output shows various log topics and their actions, such as info, error, warning, critical, and remote logs for memory, echo, and disk targets. Below this, the command `[admin@MikroTik] /system/logging> action print` is run, showing actions for memory, disk, echo, and remote targets. To the left of the terminal, a network diagram illustrates the flow of logs: "ubuntu-agent" connects to "router-agent" via "ether1", which then connects to "NAT1" via "ether2". "NAT1" has an interface "nato". The "router-agent" node is highlighted in blue. A "Topology Summary" pane on the right shows the network diagram with "telnet loc" listed under each node. A "Servers Summary" pane at the bottom right shows "lenovo CPU 3.9%, RAM 68.5%".

- Then `mikrotik-wazuh-adapter` accumulates these logs in the file `/var/log/mikrotik.log` with the help of `rsyslog`.

```

ubuntu@ubuntu-cloud:~$ cd /var/log
ubuntu@ubuntu-cloud:/var/log$ ls
README      cloud-init-output.log  dpkg.log    private
alternatives.log  cloud-init.log   journal     syslog
apport.log    dist-upgrade     kern.log    sysstat
apt          dmseg           landscape  unattended-upgrades
auth.log     dmseg.0         lastlog    wtmp
btmp        dmseg.1.gz       mikrotik.log
ubuntu@ubuntu-cloud:/var/log$ cat mikrotik.log
2025-03-06T19:44:59+00:00 MikroTik log action changed by x11-ubuntu-linux-x86-64-rv:136.0/web:admin@192.168.122.1 (/system logging action set "" b
sd-syslog=yes remote=10.0.0.2 remote-port=514 src-address=0.0.0.0 syslog-facility=daemon syslog-severity=emergency target=remote)
2025-03-06T19:45:00+00:00 MikroTik log action changed by x11-ubuntu-linux-x86-64-rv:136.0/web:admin@192.168.122.1 (/system logging action set "" b
sd-syslog=yes remote=10.0.0.2 remote-port=514 src-address=0.0.0.0 syslog-facility=daemon syslog-severity=emergency target=remote)
2025-03-06T20:45:03+00:00 MikroTik user admin logged out from 192.168.122.1 via web
2025-03-06T20:45:03+00:00 MikroTik user admin logged out from 192.168.122.1 via web
2025-03-06T20:54:11+00:00 MikroTik user admin logged in via local
2025-03-06T20:54:15+00:00 MikroTik user admin logged in from 192.168.122.1 via web
2025-03-06T20:55:06+00:00 MikroTik user admin logged out from 192.168.122.1 via web
2025-03-06T21:02:13+00:00 MikroTik ip service changed by console:admin@ttyS0/action:7 (/ip service set ssh disabled=no)
2025-03-06T21:02:53+00:00 MikroTik user admin logged in from 192.168.122.1 via ssh
2025-03-06T21:13:38+00:00 MikroTik address added by ssh:admin@192.168.122.1 (*4 = /ip address add address=1.0.0.0/21 interface=ether1)
2025-03-06T21:15:25+00:00 MikroTik address removed by ssh:admin@192.168.122.1/action:8 (/ip address remove *4)
2025-03-10T02:37:35+00:00 MikroTik router was rebooted without proper shutdown
2025-03-10T02:37:35+00:00 MikroTik to link up
2025-03-10T02:37:39+00:00 MikroTik dhcp-client on ether2 got IP address 192.168.122.107
ubuntu@ubuntu-cloud:/var/log$ 

```

- Afterwards wazuh-agent software for Ubuntu sends this logs to the Wazuh Manager Server.

```

ubuntu@ubuntu-cloud:~$ systemctl status wazuh-agent.service
● wazuh-agent.service - Wazuh agent
   Loaded: loaded (/usr/lib/systemd/system/wazuh-agent.service; enabled; pres>
   Active: active (running) since Tue 2025-03-18 10:30:15 UTC; 3h 46min ago
     Invocation: 79ed47322eb14d2fafe8be274a1da01
    Process: 669 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (cod>
      Tasks: 31 (limit: 1110)
     Memory: 326.9M (peak: 332.5M)
        CPU: 27.954s
      CGroup: /system.slice/wazuh-agent.service
          ├─883 /var/ossec/bin/wazuh-execd
          ├─921 /var/ossec/bin/wazuh-agentd
          ├─952 /var/ossec/bin/wazuh-syscheckd
          ├─966 /var/ossec/bin/wazuh-logcollector
          └─984 /var/ossec/bin/wazuh-modulesd

Notice: journal has been rotated since unit was started, output may be incomplete.
lines 1-16/16 (END)

```

- Wazuh Manager applies custom mikrotik decoders to parse necessary fields from the log message

```

nafikov@wazuh-server-ubuntu:~/wazuh-docker/single-node$ docker exec single-node-wazuh.manager-1 cat /var/ossec/etc/decoders/mikrotik_decoders.xml
<decoder name="mikrotik">
  <parent>mikrotik</parent>
  <prematch><RouterOS>1-logs: </prematch>
</decoder>

<decoder name="mikrotik1">
  <parent>mikrotik</parent>
  <regex type="pre2">\S+ (\d\d\d\d-\d\d-\d\d+\d\d+\d\d:\d\d:\d\d+\d\d:\d\d) MikroTik user (\S+) (.*)? from (\d+.+\d+.+\d+) via (\w+)</regex>
  <order>logtimestamp, logged_user, action, ip_address, protocol</order>
</decoder>

<decoder name="mikrotik1">
  <parent>mikrotik</parent>
  <regex type="pre2">\S+ (\d\d\d\d-\d\d-\d\d-\d\d+\d\d:\d\d:\d\d+\d\d:\d\d) MikroTik dhcp-client on (\S+) (.*)? address (\d+.+\d+.+\d+.+\d+)</regex>
  <order>logtimestamp, interface, action, ip_address</order>
</decoder>

<decoder name="mikrotik1">
  <parent>mikrotik</parent>
  <regex type="pre2">\S+ (\d\d\d\d-\d\d-\d\d-\d\d+\d\d:\d\d:\d\d+\d\d:\d\d) MikroTik router (\S+)</regex>
  <order>logtimestamp, action</order>
</decoder>

```

- After decoding Wazuh Manager applied custom mikrotik rules to transform raw logs to the Wazuh events

```

nafikov@wazuh-server-ubuntu:~/wazuh-docker/single-node$ docker exec single-node-wazuh.manager-1 cat /var/ossec/etc/rules/mikrotik_rules.xml
<group name="Mikrotik,>

<rule id="110000" level="0">
  <decoded_as>mikrotik</decoded_as>
  <description>Mikrotik-Event</description>
</rule>

<rule id="110001" level="5">
  <if_sid>110000</if_sid>
  <match>dhcp-client on ether</match>
  <description>MikroTik dhcp-client received an IP address ${ip_address}</description>
</rule>

<rule id="110002" level="5">
  <if_sid>110000</if_sid>
  <match>rebooted</match>
  <description>MikroTik router rebooted</description>
</rule>

<rule id="110003" level="5">
  <if_sid>110000</if_sid>
  <match>logged out from</match>
  <description>MikroTik user logged out via ${protocol}</description>
</rule>

<rule id="110004" level="5">
  <if_sid>110000</if_sid>
  <match>logged in from</match>
  <description>MikroTik user logged in from ${ip_address} via ${protocol}</description>
</rule>

</group>nafikov@wazuh-server-ubuntu:~/wazuh-docker/single-node$
```

- Finally, these logs become visible as Wazuh triggered rules

Expanded document

View surrounding documents [View single document](#)

Table	JSON
	{ "_index": "wazuh-alerts-4.x-2025.03.18" }
t agent.id	003
t agent.ip	10.0.0.2
t agent.name	mikrotik-agent
t data.action	logged in
t data.ip_address	192.168.0.2
t data.logged_user	admin
t data.logtimestamp	2025-03-18T13:50:29+00:00
t data.protocol	ssh
t decoder.name	mikrotik
t full_log	RouterOS7.1-logs: 2025-03-18T13:50:29+00:00 MikroTik user admin logged in from 192.168.0.2 via ssh
t id	1742306578.224867
t input.type	log
t location	/var/log/mikrotik.log
t manager.name	wazuh.manager
t rule.description	MikroTik user logged in from 192.168.0.2 via ssh
# rule.firetimes	2
t rule.groups	Mikrotik
t rule.id	110004
# rule.level	5
o rule.mail	false
timestamp	Mar 18, 2025 @ 17:02:58.139

## PAM: Login session opened.

- Explanation:** This log tells that someone logged in to the ubuntu-agent's system successfully.
- Why and how I am able to view it:**

- Logs come from /var/log/auth.logs

```

ubuntu@ubuntu-cloud:/var/log$ 
2025-03-18T13:55:01.870664+00:00 ubuntu-cloud CRON[2635]: pam_unix(cron:session)
: session opened for user root(uid=0) by root(uid=0)
2025-03-18T13:55:01.877275+00:00 ubuntu-cloud CRON[2635]: pam_unix(cron:session)
: session closed for user root
2025-03-18T14:05:01.889062+00:00 ubuntu-cloud CRON[2654]: pam_unix(cron:session)
: session opened for user root(uid=0) by root(uid=0)
2025-03-18T14:05:01.895555+00:00 ubuntu-cloud CRON[2654]: pam_unix(cron:session)
: session closed for user root
2025-03-18T14:15:01.906610+00:00 ubuntu-cloud CRON[2682]: pam_unix(cron:session)
: session opened for user root(uid=0) by root(uid=0)
2025-03-18T14:15:01.911709+00:00 ubuntu-cloud CRON[2682]: pam_unix(cron:session)
: session closed for user root
2025-03-18T14:17:01.920891+00:00 ubuntu-cloud CRON[2686]: pam_unix(cron:session)
: session opened for user root(uid=0) by root(uid=0)
2025-03-18T14:17:01.981400+00:00 ubuntu-cloud CRON[2686]: pam_unix(cron:session)
: session closed for user root
2025-03-18T14:25:01.938916+00:00 ubuntu-cloud CRON[2714]: pam_unix(cron:session)
: session opened for user root(uid=0) by root(uid=0)
2025-03-18T14:25:01.951604+00:00 ubuntu-cloud CRON[2714]: pam_unix(cron:session)
: session closed for user root
~
```

- Then they are sent to the Wazuh Manager Server using the wazuh-agent software

```

ubuntu@ubuntu-cloud:~$ systemctl status wazuh-agent.service
● wazuh-agent.service - Wazuh agent
   Loaded: loaded (/usr/lib/systemd/system/wazuh-agent.service; enabled; pres>
   Active: active (running) since Tue 2025-03-18 10:30:12 UTC; 4h 3min ago
     Invocation: 96c4e61c8a044ec4818b006a941e5e17
    Process: 679 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (cod>
      Tasks: 31 (limit: 1058)
     Memory: 227.6M (peak: 303.4M)
        CPU: 29.141s
       CGroup: /system.slice/wazuh-agent.service
               ├─842 /var/ossec/bin/wazuh-execd
               ├─858 /var/ossec/bin/wazuh-agentd
               ├─890 /var/ossec/bin/wazuh-syscheckd
               ├─924 /var/ossec/bin/wazuh-logcollector...
               └─940 /var/ossec/bin/wazuh-modulesd

Mar 18 10:30:06 ubuntu-cloud env[679]: Deleting PID file '/var/ossec/var/run/wa>
Mar 18 10:30:06 ubuntu-cloud env[679]: Deleting PID file '/var/ossec/var/run/wa>
Mar 18 10:30:06 ubuntu-cloud env[679]: Deleting PID file '/var/ossec/var/run/wa>
Mar 18 10:30:06 ubuntu-cloud env[679]: Started wazuh-execd...
Mar 18 10:30:07 ubuntu-cloud env[679]: Started wazuh-agentd...
Mar 18 10:30:08 ubuntu-cloud env[679]: Started wazuh-syscheckd...
Mar 18 10:30:09 ubuntu-cloud env[679]: Started wazuh-logcollector...
Mar 18 10:30:10 ubuntu-cloud env[679]: Started wazuh-modulesd...
lines 1-23
```

- Then they are parsed using default decoders and default rules are applied
- Finally, we can see them in the Wazuh Dashboard interface

Expanded document		<a href="#">View surrounding documents</a>	<a href="#">View single document</a>
<a href="#">Table</a> <a href="#">JSON</a>			
t _index	wazuh-alerts-4.x-2025.03.18		
t agent.id	002		
t agent.ip	192.168.0.2		
t agent.name	ubuntu-agent		
t data.dstuser	ubuntu(uid=1000)		
t data.srcuser	ubuntu		
t data.uid	0		
t decoder.name	pam		
t decoder.parent	pam		
t full_log	Mar 18 11:06:43 ubuntu-cloud (systemd)[1682]: pam_unix(systemd-user:session): session opened for user ubuntu(uid=1000) by ubuntu(uid=0)		
t id	1742296004.221089		
t input.type	log		
t location	journald		
t manager.name	wazuh.manager		
t predecoder.hostname	ubuntu-cloud		
t predecoder.program_name	(systemd)		
t predecoder.timestamp	Mar 18 11:06:43		
t rule.description	PAM: Login session opened.		
# rule.firetimes	2		
t rule.gdpr	IV_32.2		
t rule.gpg13	7.8, 7.9		
t rule.groups	pam, syslog, authentication_success		
t rule.hipaa	164.312.b		
t rule.id	5501		
# rule.level	3		

## Task 3 - Use cases

a.

## Task description

- a. Demonstrate how to block malicious IP addresses from accessing web resources on a web server. To do this, you will set up your web servers on select endpoints within your infrastructure, and try to access them from an external endpoint.

I used the [following PoC guide](#) from the official Wazuh website

# Set up web server

- I decided to use `ubuntu-agent` machine as a web server
  - I installed NGINX on the `ubuntu-agent` machine

```
ubuntu@ubuntu-cloud:~$ apt show nginx
Package: nginx
Version: 1.26.0-2ubuntu3.2
Priority: optional
Section: web
Origin: Ubuntu
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Debian Nginx Maintainers <pkg-nginx-maintainers@alioth-lists.debian.net>
Bugs: https://bugs.launchpad.net/ubuntu/+filebug
Installed-Size: 1574 kB
Provides: httpd, httpd-cgi, nginx-abi-1.26.0-1
Depends: libc6 (>= 2.34), libcrypt1 (>= 1:4.1.0), libpcre2-8-0 (>= 10.22), libssl3t64 (>= 3.0.0), zlib1g (>= 1:1.1.4), iproute2, nginx-common (= 1.26.0-2ubuntu3.2)
Breaks: nginx-core (<< 1.22.1-6~), nginx-extras (<< 1.22.1-6~), nginx-light (<< 1.22.1-6~)
Replaces: nginx-core (<< 1.22.1-6~), nginx-extras (<< 1.22.1-6~), nginx-light (<< 1.22.1-6~)
Homepage: https://nginx.org
Download-Size: 599 kB
APT-Manual-Installed: yes
APT-Sources: http://archive.ubuntu.com/ubuntu oracular-updates/main amd64 Packages
Description: small, powerful, scalable web/proxy server
Nginx ("engine X") is a high-performance web and reverse proxy server
created by Igor Sysoev. It can be used both as a standalone web server
and as a proxy to reduce the load on back-end HTTP or mail servers.
```

- And checked if it works

ubuntu@ubuntu-cloud:~\$ systemctl status nginx.service

● nginx.service - A high performance web server and a reverse proxy server

  Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)

  Active: active (running) since Tue 2025-03-18 14:44:49 UTC; 4min 41s ago

  Invocation-Id: fcb4abd50222427cac880de0b2882dda

    Docs: man:nginx(8)

  Process: 3063 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master\_process

  Process: 3072 ExecStart=/usr/sbin/nginx -g daemon on; master\_process on; (c)

  Main PID: 3104 (nginx)

    Tasks: 2 (limit: 1058)

    Memory: 2.1M (peak: 4.7M)

    CPU: 40ms

    CGroup: /system.slice/nginx.service

          └─3104 "nginx: master process /usr/sbin/nginx -g daemon on; master\_

              └─3107 "nginx: worker process"

Mar 18 14:44:48 ubuntu-cloud systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...

Mar 18 14:44:49 ubuntu-cloud systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server...

- Finally, I was able to access NGINX main page on 80 port

```
ubuntu@ubuntu-cloud:~$ curl localhost
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
ubuntu@ubuntu-cloud:~$
```

## Access web server from malicious IP address

- I decided to make `mikrotik-wazuh-adapter` machine malicious because it should not check my `ubuntu-agent` machine that is connected to the same router. Adapter's main task is just to transfer logs from the `router-agent` to the Wazuh Manager Server.
- IP address of the `ubuntu-agent` is `192.168.0.2/24`

```
ubuntu@ubuntu-cloud:~$ ip a | grep inet
inet 127.0.0.1/8 scope host lo
inet6 ::1/128 scope host noprefixroute
inet 192.168.0.2/24 brd 192.168.0.255 scope global e0
inet6 fe80::eef:fe5a:0/64 scope link proto kernel ll
ubuntu@ubuntu-cloud:~$
```

- I was able to access the web server from the `mikrotik-wazuh-adapter` (`10.0.0.2/24`)

```
ubuntu@ubuntu-cloud:~$ ip a | grep inet
inet 127.0.0.1/8 scope host lo
inet6 ::1/128 scope host noprefixroute
inet 10.0.0.2/24 brd 10.0.0.255 scope global e0
inet6 fe80::e19:3bff:feff:0/64 scope link proto kernel_ll
ubuntu@ubuntu-cloud:~$ curl 192.168.0.2
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
ubuntu@ubuntu-cloud:~$
```

- And we can see in the NGINX access logs that `10.0.0.2` accessed the web server on `ubuntu-agent` machine

```

ubuntu@ubuntu-cloud:/var/log/nginx$ cat access.log
::1 - - [18/Mar/2025:14:45:55 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/8.9.1"
10.0.0.2 - - [18/Mar/2025:14:46:52 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/8.9.1"
::1 - - [18/Mar/2025:14:50:21 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/8.9.1"
10.0.0.2 - - [18/Mar/2025:14:54:59 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/8.9.1"
ubuntu@ubuntu-cloud:/var/log/nginx$

```

## Send NGINX access logs

- To block IP to access the web server we firstly need to recognise it, so Wazuh should have NGINX accesslogs
- To configure NGINX access logs to be sent to the Wazuh Manger Server I added `/var/log/nginx/access.log` as a local source of logs in the `var/ossec/etc/ossec.conf`

```

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

```

```

ubuntu@ubuntu-cloud:~$ sudo cat /var/ossec/etc/ossec.conf | grep nginx -B 20 -A 15
ginx -B 20 -A 15
</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>
  <localfile>
    <log_format>syslog</log_format>
    <location>/var/log/nginx/access.log</location>
  </localfile>
</ossec_config>
ubuntu@ubuntu-cloud:~$ 

```

- Then I need to restart the wazuh-agent service

```

ubuntu@ubuntu-cloud:~$ sudo systemctl restart 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; pres
  Active: active (running) since Tue 2025-03-18 15:14:34 UTC; 18s ago
  Invocation: 82c5eb91b61d4aa9f95efcc841b75396e
  Process: 4105 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (co
    Tasks: 32 (limit: 1058)
  Memory: 18.5M (peak: 20.6M)
    CPU: 9.533s
   CGroup: /system.slice/wazuh-agent.service
           └─4127 /var/ossec/bin/wazuh-execd
             ├─4138 /var/ossec/bin/wazuh-agentd
             ├─4152 /var/ossec/bin/wazuh-syscheckd
             ├─4163 /var/ossec/bin/wazuh-logcollector
             ├─4180 /var/ossec/bin/wazuh-modulesd

Mar 18 15:14:28 ubuntu-cloud systemd[1]: Starting wazuh-agent.service - Wazuh a...
Mar 18 15:14:28 ubuntu-cloud env[4105]: Starting Wazuh v4.11.0...
Mar 18 15:14:29 ubuntu-cloud env[4105]: Started wazuh-execd...
Mar 18 15:14:30 ubuntu-cloud env[4105]: Started wazuh-agentd...
Mar 18 15:14:30 ubuntu-cloud env[4105]: Started wazuh-syscheckd...
Mar 18 15:14:31 ubuntu-cloud env[4105]: Started wazuh-logcollector...
Mar 18 15:14:32 ubuntu-cloud env[4105]: Started wazuh-modulesd...
Mar 18 15:14:34 ubuntu-cloud env[4105]: Completed.
lines 1-23

```

# Block malicious IP address

- After things done I connected back to the Wazuh Manager Server

```
~ nafikov@wazuh-server-ubuntu:~ ssh -i .ssh/wazuh_server nafikov@89.169.173.179
nafikov@wazuh-server-ubuntu:~$ docker exec -it single-node-wazuh.manager-1 bash
bash-5.2#
```

- I installed wget using yum package manager

```
~ nafikov@wazuh-server-ubuntu:~ ssh -i .ssh/wazuh_server nafikov@89.169.173.179
bash-5.2# yum update
Last metadata expiration check: 1:32:06 ago on Tue Mar 18 14:03:03 2025.
Dependencies resolved.
Nothing to do.
Complete!
bash-5.2# yum install -y wget
Last metadata expiration check: 1:32:16 ago on Tue Mar 18 14:03:03 2025.
Dependencies resolved.

=====
Package           Architecture      Version       Repository   Size
=====
Installing:
  wget            x86_64          1.21.3-1.amzn2023.0.4
Installing dependencies:
  gnutls          x86_64          3.8.0-381.amzn2023.0.7
  libmetalink    x86_64          0.1.3-14.amzn2023.0.2
  nettle          x86_64          3.8-1.amzn2023.0.2

Transaction Summary
=====
Install 4 Packages

Total download size: 2.3 M
Installed size: 7.2 M
Downloading Packages:
(1/4): libmetalink-0.1.3-14.amzn2023.0.2.x86_64.rpm          95 kB/s | 31 kB   00:00
(2/4): gnutls-3.8.0-381.amzn2023.0.7.x86_64.rpm             2.2 MB/s | 1.1 MB  00:00
(3/4): nettle-3.8-1.amzn2023.0.2.x86_64.rpm                799 kB/s | 417 kB  00:00
(4/4): wget-1.21.3-1.amzn2023.0.4.x86_64.rpm              3.7 MB/s | 779 kB  00:00

Total                                         1.7 MB/s | 2.3 MB  00:01

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing:                               1/1
  Installing : nettle-3.8-1.amzn2023.0.2.x86_64          1/4
  Installing : gnutls-3.8.0-381.amzn2023.0.7.x86_64        2/4
  Installing : libmetalink-0.1.3-14.amzn2023.0.2.x86_64    3/4
  Installing : wget-1.21.3-1.amzn2023.0.4.x86_64          4/4
  Running scriptlet: wget-1.21.3-1.amzn2023.0.4.x86_64    4/4
  Verifying   : gnutls-3.8.0-381.amzn2023.0.7.x86_64        1/4
  Verifying   : libmetalink-0.1.3-14.amzn2023.0.2.x86_64    2/4
  Verifying   : nettle-3.8-1.amzn2023.0.2.x86_64          3/4
  Verifying   : wget-1.21.3-1.amzn2023.0.4.x86_64          4/4

Installed:
  gnutls-3.8.0-381.amzn2023.0.7.x86_64      libmetalink-0.1.3-14.amzn2023.0.2.x86_64      nettle-3.8-1.amzn2023.0.2.x86_64      wget-1.21.3-1.amzn2023.0.4.x86_64

Complete!
bash-5.2#
```

- I downloaded [AlienVault IP reputation database](https://github.com/OSSEC/OSSEC-IDS/tree/master/lists/alienVault_reputation.ipset) from GitHub:

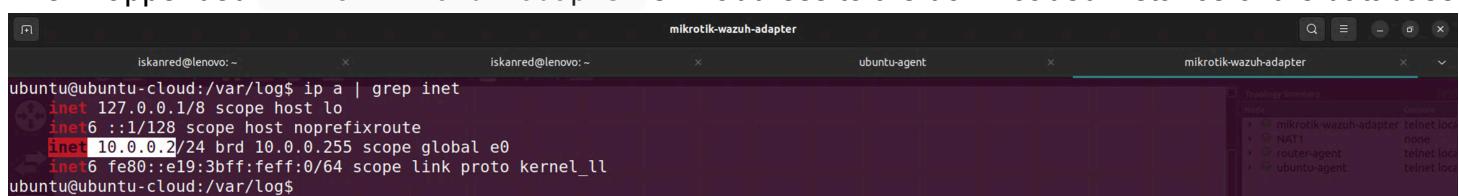
```
wget https://raw.githubusercontent.com/firehol/blocklist-ipsets/master/alienvault_reputation.ipset -O
/var/ossec/etc/lists/alienvault_reputation.ipset
```

```
~ nafikov@wazuh-server-ubuntu:~ ssh -i .ssh/wazuh_server nafikov@89.169.173.179
bash-5.2# wget https://raw.githubusercontent.com/firehol/blocklist-ipsets/master/alienvault_reputation.ipset -O /var/ossec/etc/lists/alienvault_reputation.ipset
--2025-03-18 15:40:43-- https://raw.githubusercontent.com/firehol/blocklist-ipsets/master/alienvault_reputation.ipset
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133, 185.199.108.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9495 (9.3K) [text/plain]
Saving to: '/var/ossec/etc/lists/alienvault_reputation.ipset'

/var/ossec/etc/lists/alienvault_reputation.ipset 100%[=====] 9.27K ---KB/s in 0.003s

2025-03-18 15:40:44 (2.67 MB/s) - '/var/ossec/etc/lists/alienvault_reputation.ipset' saved [9495/9495]
```

- Then I appended mikrotik-wazuh-adapter's IP address to the downloaded instance of the database



```
mikrotik-wazuh-adapter
ubuntu@ubuntu-cloud:~ ssh -i .ssh/wazuh_server nafikov@89.169.173.179
ubuntu@ubuntu-cloud:~ var/log$ ip a | grep inet
inet 127.0.0.1/8 scope host lo
inet6 ::1/128 scope host noprefixroute
inet 10.0.0.2/24 brd 10.0.0.255 scope global e0
inet6 fe80::e19:3bff:fe:0/64 scope link proto kernel ll
ubuntu@ubuntu-cloud:~ var/log$
```

```
echo "10.0.0.2" >> /var/ossec/etc/lists/alienVault_reputation.ipset
```

```
~ nafikov@wazuh-server-ubuntu: ~ ssh -i .ssh/wazuh_server nafikov@89.169.173.179
bash-5.2# echo "10.0.0.2" >> /var/ossec/etc/lists/alienVault_reputation.ipset
bash-5.2#
```

- Afterwards, I downloaded a [script](#) to convert the database from `.ipset` to the `.cdb` format from the Wazuh official website.

```
wget https://wazuh.com/resources/iplist-to-cdblist.py -O /tmp/iplist-to-cdblist.py
```

```
~ nafikov@wazuh-server-ubuntu: ~ ssh -i .ssh/wazuh_server nafikov@89.169.173.179
bash-5.2# wget https://wazuh.com/resources/iplist-to-cdblist.py -O /tmp/iplist-to-cdblist.py
--2025-03-18 15:45:46-- https://wazuh.com/resources/iplist-to-cdblist.py
Resolving wazuh.com (wazuh.com)... 18.165.183.42, 18.165.183.38, 18.165.183.6, ...
Connecting to wazuh.com (wazuh.com)18.165.183.42:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1570 (1.5K) [binary/octet-stream]
Saving to: '/tmp/iplist-to-cdblist.py'

/tmp/iplist-to-cdblist.py          100%[=====]  1.53K --.-KB/s   in 0s
2025-03-18 15:45:47 (18.3 MB/s) - '/tmp/iplist-to-cdblist.py' saved [1570/1570]
bash-5.2#
```

- And converted the database instance using the script

```
/var/ossec/framework/python/bin/python3 /tmp/iplist-to-cdblist.py
/var/ossec/etc/lists/alienVault_reputation.ipset /var/ossec/etc/lists/blacklist-
alienVault
```

```
~ nafikov@wazuh-server-ubuntu: ~ ssh -i .ssh/wazuh_server nafikov@89.169.173.179
bash-5.2# /var/ossec/framework/python/bin/python3 /tmp/iplist-to-cdblist.py /var/ossec/etc/lists/alienVault_reputation.ipset /var/ossec/etc/lists/blacklist-alienVault
[/var/ossec/etc/lists/alienVault_reputation.ipset] -> [/var/ossec/etc/lists/blacklist-alienVault]
bash-5.2#
```

- Also, I removed the original IPSET database and the script since they are unnecessary now

```
~ nafikov@wazuh-server-ubuntu: ~ ssh -i .ssh/wazuh_server nafikov@89.169.173.179
bash-5.2# rm -rf /var/ossec/etc/lists/alienVault_reputation.ipset
bash-5.2# rm -rf /tmp/iplist-to-cdblist.py
bash-5.2#
```

- Then I assigned the right permissions to the generated file for the `wazuh` user

```
~ nafikov@wazuh-server-ubuntu: ~ ssh -i .ssh/wazuh_server nafikov@89.169.173.179
bash-5.2# chown wazuh:wazuh /var/ossec/etc/lists/blacklist-alienVault
bash-5.2# ls -l /var/ossec/etc/lists/
total 36
drwxrwx--- 2 wazuh wazuh 4096 Mar  3 19:05 amazon
-rw-rw---- 1 wazuh wazuh 107 Feb 14 14:10 audit-keys
-rw-rw---- 1 wazuh wazuh 2265 Mar  3 19:05 audit-keys.cdb
-rw-r--r-- 1 wazuh wazuh 9315 Mar 18 15:48 blacklist-alienVault
-rw-rw---- 1 wazuh wazuh 892 Feb 14 14:10 security-eventchannel
-rw-rw---- 1 wazuh wazuh 6461 Mar  3 19:05 security-eventchannel.cdb
bash-5.2#
bash-5.2#
```

- I added a new custom rule to trigger Wazuh active response script.

```
vim /var/ossec/etc/rules/local_rules.xml
```

```
<group name="attack">
  <rule id="100100" level="10">
    <if_group>web|attack|attacks</if_group>
    <list field="srcip" lookup="address_match_key">etc/lists/blacklist-
alienVault</list>
    <description>IP address found in AlienVault reputation database.</description>
```

```
</rule>  
</group>
```

- I added `etc/lists/blacklist-alienVault` **list** entity from this rule to the default ruleset inside the `/var/ossec/etc/ossec.conf`

```
<ruleset>
...
<list>etc/lists/blacklist-alienVault
...
</ruleset>
```

```
~ -- nafikov@wazuh-server-ubuntu: ~ -- ssh -i .ssh/wazuh_server nafikov@89.169.173.17
bash-5.2# cat /var/ossec/etc/ossec.conf | grep "<list>etc/lists/blacklist-alienVault</list>" -B 8 -A 12
<ruleset>
  <!-- Default ruleset -->
  <decoder_dir>ruleset/decoders</decoder_dir>
  <rule_dir>ruleset/rules</rule_dir>
  <rule_exclude>0215-policy_rules.xml</rule_exclude>
  <list>etc/lists/audit-keys</list>
  <list>etc/lists/amazon/aws-eventnames</list>
  <list>etc/lists/security-eventchannel</list>
  <list>etc/lists/blacklist-alienVault</list>

  <!-- User-defined ruleset -->
  <decoder_dir>etc/decoders</decoder_dir>
  <rule_dir>etc/rules</rule_dir>
</ruleset>

<rule_test>
  <enabled>yes</enabled>
  <threads>1</threads>
  <max_sessions>64</max_sessions>
  <session_timeout>15m</session_timeout>
</rule_test>
bash-5.2#
```

- Finally, I added Active Response block to the Wazuh server `/var/ossec/etc/ossec.conf` file that performs `firewall-drop` which integrates with the Ubuntu local `iptables` firewall and drop incoming network connection from the attacker endpoint for the given timeout:

```

<ossec_config>
...
<active-response>
    <command>firewall-drop</command>    <!-- Drop via iptables firewall -->
    <location>local</location>
    <rules_id>100100</rules_id>
    <timeout>60</timeout>                  <!-- Block attacker's traffic for 60 seconds --
->
</active-response>
...
</ossec_config>

```

```

bash-5.2# cat /var/ossec/etc/ossec.conf | grep "<active-response>" -B 47 -A 5
~ -- nafikov@wazuh-server-ubuntu: ~ -- ssh -i .ssh/wazuh_server nafikov@89.169.173.179
<!-- Active response -->
<global>
    <white_list>127.0.0.1</white_list>
    <white_list>^localhost.localdomain$</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>host-deny</name>
    <executable>host-deny</executable>
    <timeout_allowed>yes</timeout_allowed>
</command>

<command>
    <name>route-null</name>
    <executable>route-null</executable>
    <timeout_allowed>yes</timeout_allowed>
</command>

<command>
    <name>win_route-null</name>
    <executable>route-null.exe</executable>
    <timeout_allowed>yes</timeout_allowed>
</command>

<command>
    <name>netsh</name>
    <executable>netsh.exe</executable>
    <timeout_allowed>yes</timeout_allowed>
</command>

<active-response>
    <command>firewall-drop</command>
    <location>local</location>
    <rules_id>100100</rules_id>
    <timeout>60</timeout>
</active-response>
bash-5.2#

```

- And restarted the Wazuh

```

nafikov@wazuh-server-ubuntu:~/wazuh-docker/single-node$ docker compose down
WARN[0000] /home/nafikov/wazuh-docker/single-node/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 4/4
✓ Container single-node-wazuh.dashboard-1 Removed
✓ Container single-node-wazuh.manager-1 Removed
✓ Container single-node-wazuh.indexer-1 Removed
✓ Network single-node_default Removed
nafikov@wazuh-server-ubuntu:~/wazuh-docker/single-node$ docker compose up -d
WARN[0000] /home/nafikov/wazuh-docker/single-node/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 4/4
✓ Network single-node_default Created
✓ Container single-node-wazuh.indexer-1 Started
✓ Container single-node-wazuh.manager-1 Started
✓ Container single-node-wazuh.dashboard-1 Started
nafikov@wazuh-server-ubuntu:~/wazuh-docker/single-node$ 

```

## Relusts

- Let's check if things were done correctly
- Firstly, we see that now `mirkotik-wazuh-adapter` has a 60 seconds timeout after accessing the web server on `ubuntu-agent` machine.

```

ubuntu@ubuntu-cloud:/var/log$ curl 192.168.0.2 -m 10
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
ubuntu@ubuntu-cloud:/var/log$ curl 192.168.0.2 -m 10
curl: (28) Connection timed out after 10001 milliseconds
ubuntu@ubuntu-cloud:/var/log$

```

```

ubuntu@ubuntu-cloud:/var/log$ curl 192.168.0.2 -m 10
curl: (28) Connection timed out after 10003 milliseconds
ubuntu@ubuntu-cloud:/var/log$

```

- We can prove it using some script. The script below make requests every second for 61 seconds.

```
for i in {1..61}; do curl 192.168.0.2; sleep 1; done
```

- Running this script we can see that now an attacker can make at maximum 1 RPM (request per minute).

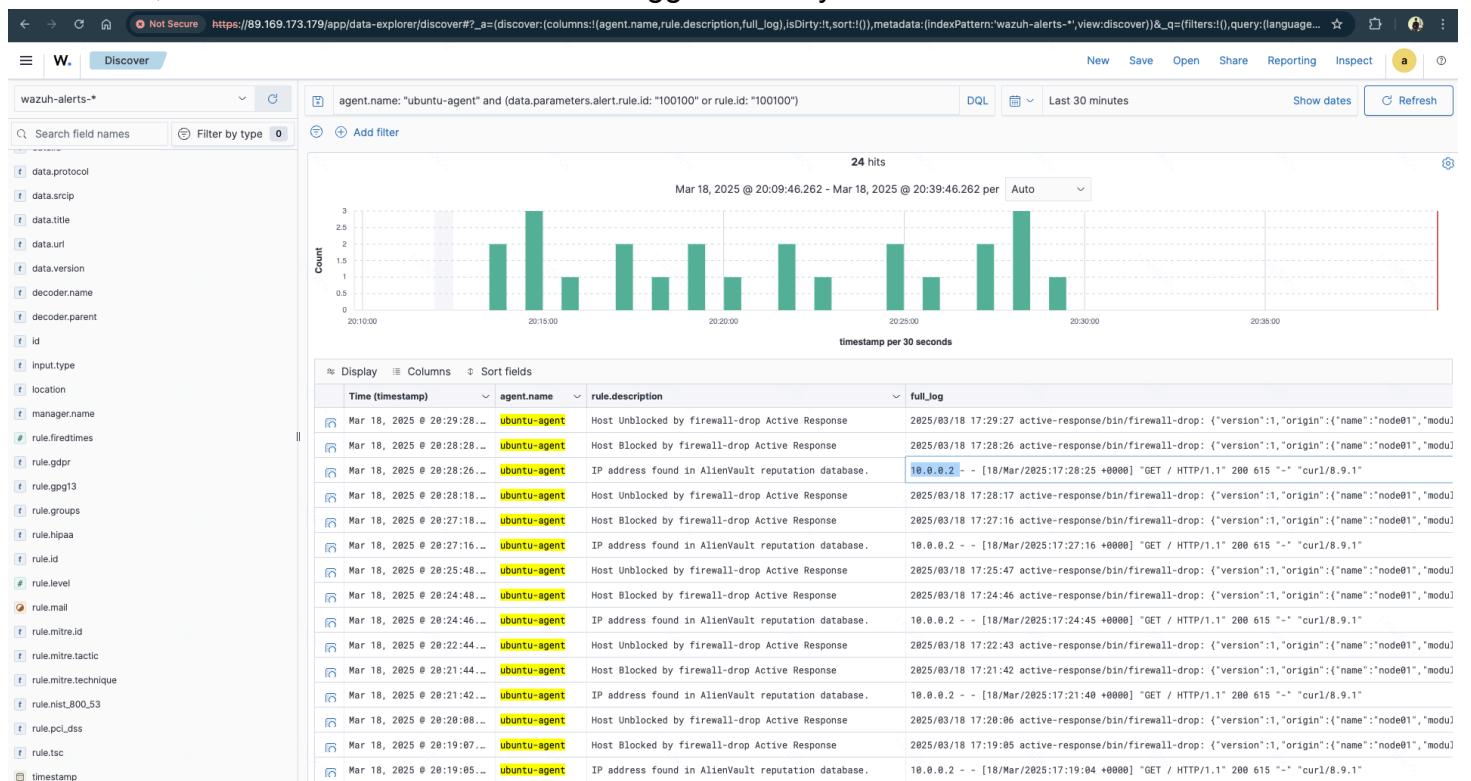
```

neuntu@ubuntu-cloud:/var/log$ for i in {1..61}; do curl 192.168.0.2; sleep 1; done
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
<p><em>Thank you for using nginx.</em></p>
</body>
</html>

```

- We see that in 61 seconds `curl` was successfully executed only 2 times: the 1st one, and the one after 60 seconds.

- In addition, now we can see the rules were triggered many times



- We can see that our `ubuntu-agent` successfully sends logs to the Wazuh Manager Server which detects IP address `10.0.0.2` which is allegedly from AlienVault reputation database and blocks requests from this address for 60 seconds. In a minute it unblocks this IP address.

b.

### Task description

- b. Simulate a brute force attack against your infrastructure and demonstrate how you would detect the attack on each of the devices within your infrastructure. Are you able to detect the attack? If not, ensure you are able to.

- For simulating a brute force attack I used [hydra](#) tool
- Let's start with trying to brute force `ubuntu-agent` and `mikrotik-agent` via SSH protocol making login requests using different credentials.

## New rule for Mikrotik

- But before I need to configure another rule for `mikrotik-agent` : **login failure**
- Let's add the following new decoders to the `var/ossec/etc/decoders/mikrotik_decoders.xml`:

```
<decoder name="mikrotik1">
  <parent>mikrotik</parent>
  <regex type="pcre2">\S+ (\d\d\d\d-\d\d-\d\d+\d\d\d:\d\d:\d\d+\+\d\d:\d\d)
MikroTik (.*) for user (\S+) from (\d+\.\d+\.\d+\.\d+) via (\w+)</regex>
  <order>logtimestamp, action, possible_user, ip_address, protocol</order>
```

```
</decoder>

<decoder name="mikrotik1">
  <parent>mikrotik</parent>
  <regex type="pcre2">\S+ (\d\d\d\d-\d\d-\d\d+\d\d+\d\d:\d\d)
MikroTik (.*) message repeated (\d+) times: \[\s*(.*) for user (\S+) from
(\d+\.\d+\.\d+\.\d+) via (\w+)\s*\]</regex>
  <order>logtimestamp, action_message, repeat_times, action, possible_user,
ip_address, protocol</order>
</decoder>
```

- And new rules to the var/ossec/etc/rules/mikrotik\_rules.xml

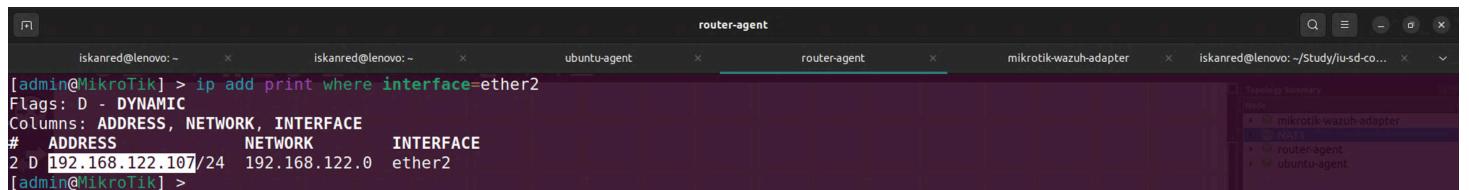
```
<rule id="110005" level="5">
  <if_sid>110000</if_sid>
  <match>login failure</match>
  <description>Someone tried to login Mikrotik user $(possible_user) from
$(ip_address) via $(protocol)</description>
</rule>

<rule id="110006" level="5">
  <if_sid>110000</if_sid>
  <field name="action_message">login</field>
  <match>failure</match>
  <description>Someone tried to login Mikrotik user $(possible_user) from
$(ip_address) via $(protocol) $(repeat_times) times</description>
</rule>
```

- And reboot the wazuh-manager

```
bash-5.2# ~ nafikov@wazuh-server-ubuntu: ~/wazuh-docker/single-node -- ssh -i .ssh/wazuh_server nafikov@89.169.173.179
bash-5.2# vim /var/ossec/etc/decoders/mikrotik_decoders.xml
bash-5.2# vim /var/ossec/etc/rules/mikrotik_rules.xml
bash-5.2# exit
exit
nafikov@wazuh-server-ubuntu:~/wazuh-docker/single-node$ docker compose restart
WARN[0000] /home/nafikov/wazuh-docker/single-node/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Restarting 3/3
✓ Container single-node-wazuh.indexer-1 Started
✓ Container single-node-wazuh.manager-1 Started
✓ Container single-node-wazuh.dashboard-1 Started
0.7s
4.4s
10.7s
```

- Now let's try to fail login to the router via SSH from my host machine:

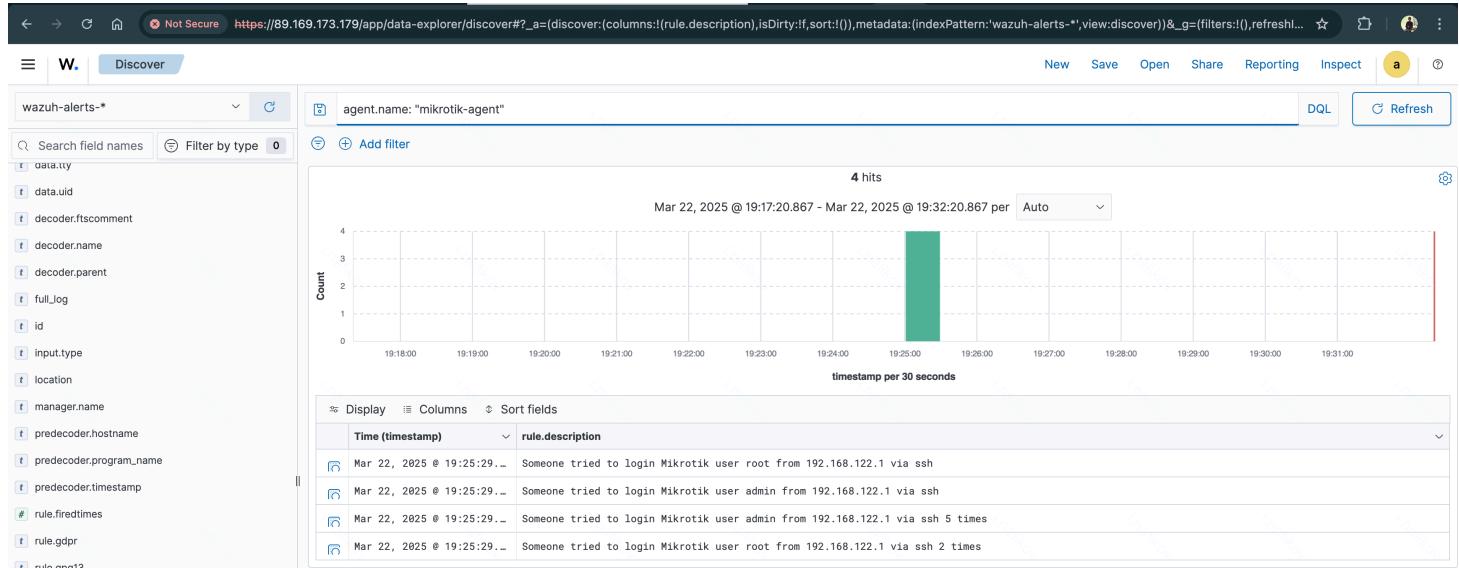


```

iskanred@lenovo: ~      iskanred@lenovo: ~      ubuntu-agent      router-agent      mikrotik-wazuh-adapter      iskanred@lenovo: ~/Study/iu-sd-course/lab-04
iskanred@lenovo:~/Study/iu-sd-course/lab-04$ ssh admin@192.168.122.107
admin@192.168.122.107's password:
Permission denied, please try again.
admin@192.168.122.107's password:
Permission denied, please try again.
admin@192.168.122.107's password:
admin@192.168.122.107: Permission denied (password).
iskanred@lenovo:~/Study/iu-sd-course/lab-04$ ssh root@192.168.122.107
root@192.168.122.107's password:
Permission denied, please try again.
root@192.168.122.107's password:
Permission denied, please try again.
root@192.168.122.107's password:
root@192.168.122.107: Permission denied (password).
iskanred@lenovo:~/Study/iu-sd-course/lab-04$ 

```

- Finally, we can see these logs on the Wazuh dashboard



## Brute force attack on Ubuntu

- I configured `hydra` to brute force SSH service with some logins provided and passwords of length 2 containing digits only

```

iskanred@lenovo: ~      iskanred@lenovo: ~      mikrotik-wazuh-adapter      iskanred@lenovo: ~/Study/iu-sd-course/lab-04      router-agent      ubuntu-agent      iskanred@lenovo: ~/Study/iu-sd-course/lab-04
iskanred@lenovo:~/Study/iu-sd-course/lab-04$ cat logins
admin
login
user
root
iskanred@lenovo:~/Study/iu-sd-course/lab-04$ 

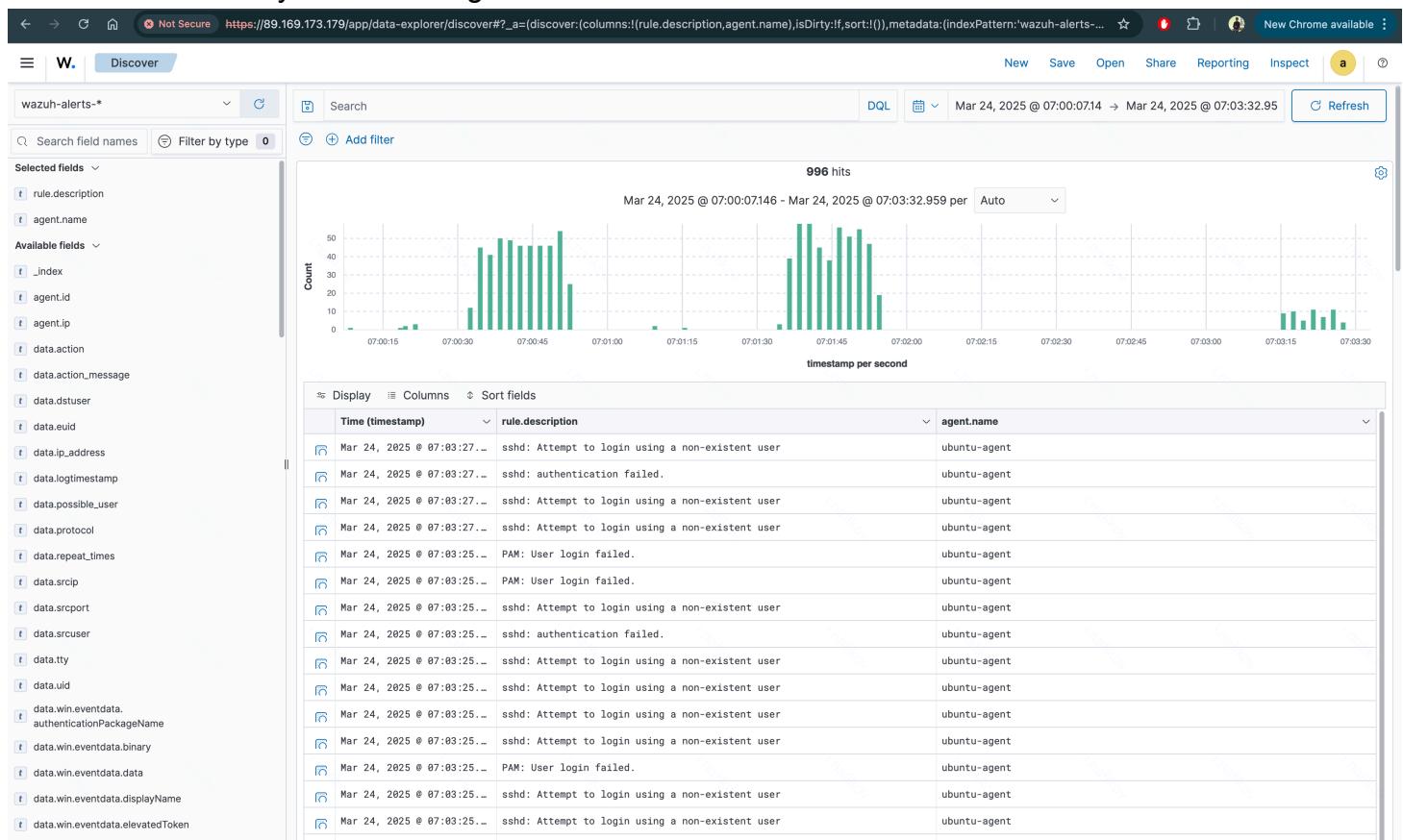
ubuntu@ubuntu-cloud:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: e0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 0c:ee:0a:5a:00:00 brd ff:ff:ff:ff:ff:ff
    altname enp0s3
    inet 192.168.0.2/24 brd 192.168.0.255 scope global e0
        valid_lft forever preferred_lft forever
    inet6 fe80::eeaf:fe5a::/64 scope link proto kernel ll
        valid_lft forever preferred_lft forever
ubuntu@ubuntu-cloud:~$ 

iskanred@lenovo: ~      iskanred@lenovo: ~      mikrotik-wazuh-adapter      iskanred@lenovo: ~/Study/iu-sd-course/lab-04      router-agent      ubuntu-agent      iskanred@lenovo: ~/Study/iu-sd-course/lab-04
iskanred@lenovo:~/Study/iu-sd-course/lab-04$ hydra -t 4 -I -L logins -x 2:2:1 192.168.0.2 ssh
Hydra v9.2 (c) 2021 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 2025-03-24 07:05:15
[WARNING] Restorefile (ignored ...) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 4 tasks per 1 server, overall 4 tasks, 400 login tries (l:4/p:100), ~100 tries per task
[DATA] attacking ssh://192.168.0.2:22/
[STATUS] 74.00 tries/min, 74 tries in 00:01h, 326 to do in 00:05h, 4 active
NAT1

```

- So I can immediately see these login tries in Wazuh:



## Brute force attack on Mikrotik

- Then I did the same to attack the router which was much faster

The terminal window shows the execution of the Hydra password cracking tool against a Mikrotik router. The command used is `hydra -t4 -I -L logins -x 2:2:1 192.168.122.107 ssh`. The output indicates that the attack started at 07:08:36 on March 24, 2025, and completed at 07:08:55. It found 0 valid passwords. The session summary shows the target configuration, including NAT1, router-agent, and ubuntu-agent services.

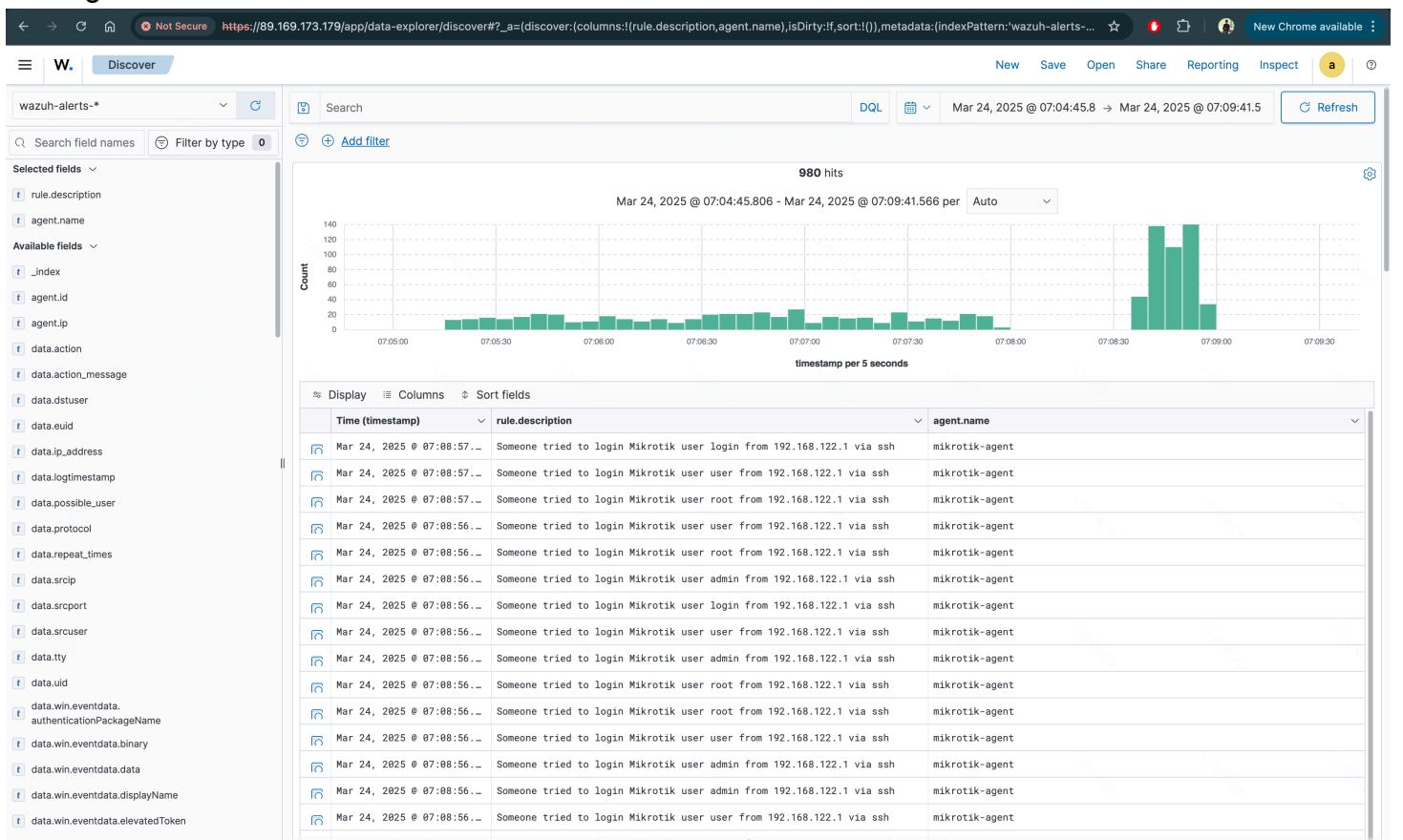
```

iskanred@lenovo:~/Study/iu-sd-course/lab-04$ hydra -t4 -I -L logins -x 2:2:1 192.168.122.107 ssh
Hydra v9.2 (c) 2021 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 2025-03-24 07:08:36
[WARNING] Restorefile (ignored ...) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 4 tasks per 1 server, overall 4 tasks, 400 login tries (l:4:p:100), ~100 tries per task
[DATA] attacking ssh://192.168.122.107:22/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-03-24 07:08:55
iskanred@lenovo:~/Study/iu-sd-course/lab-04$ 

```

- And again in Wazuh I was able to monitor the attack:



## Brute force attack on Windows

- I did exactly the same to the windows-agent

 Select Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\iskanred> ipconfig

Windows IP Configuration

Wireless LAN adapter Local Area Connection* 1:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 10:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Ethernet adapter VMware Network Adapter VMnet1:

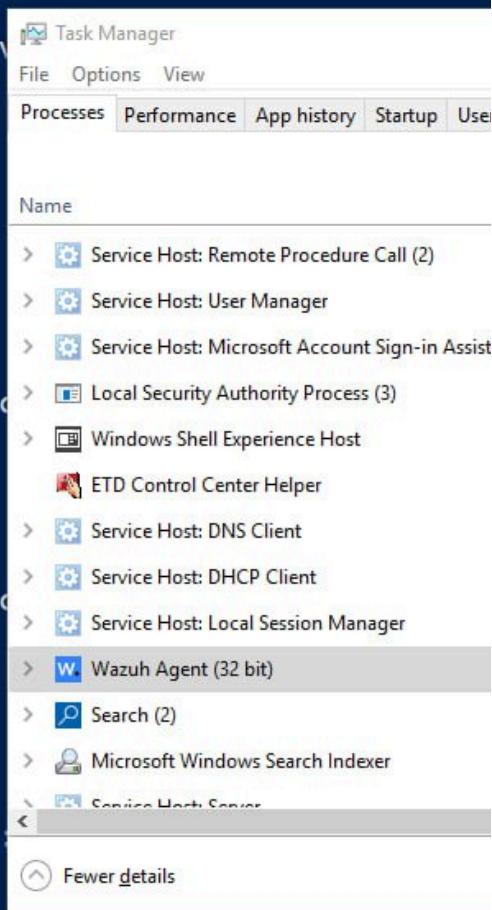
    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::c4c9:14d%
    IPv4 Address. . . . . : 192.168.197.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :

Ethernet adapter VMware Network Adapter VMnet8:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::eb9e:2d68%
    IPv4 Address. . . . . : 192.168.44.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :

Wireless LAN adapter Wi-Fi:

    Connection-specific DNS Suffix  . :
    IPv6 Address. . . . . : 2a00:1fa1:821e:
    Temporary IPv6 Address. . . . . : 2a00:1fa1:821e:
    Link-local IPv6 Address . . . . . : fe80::eeaf:3481%
    IPv4 Address. . . . . : 172.20.10.7
    Subnet Mask . . . . . : 255.255.255.240
```

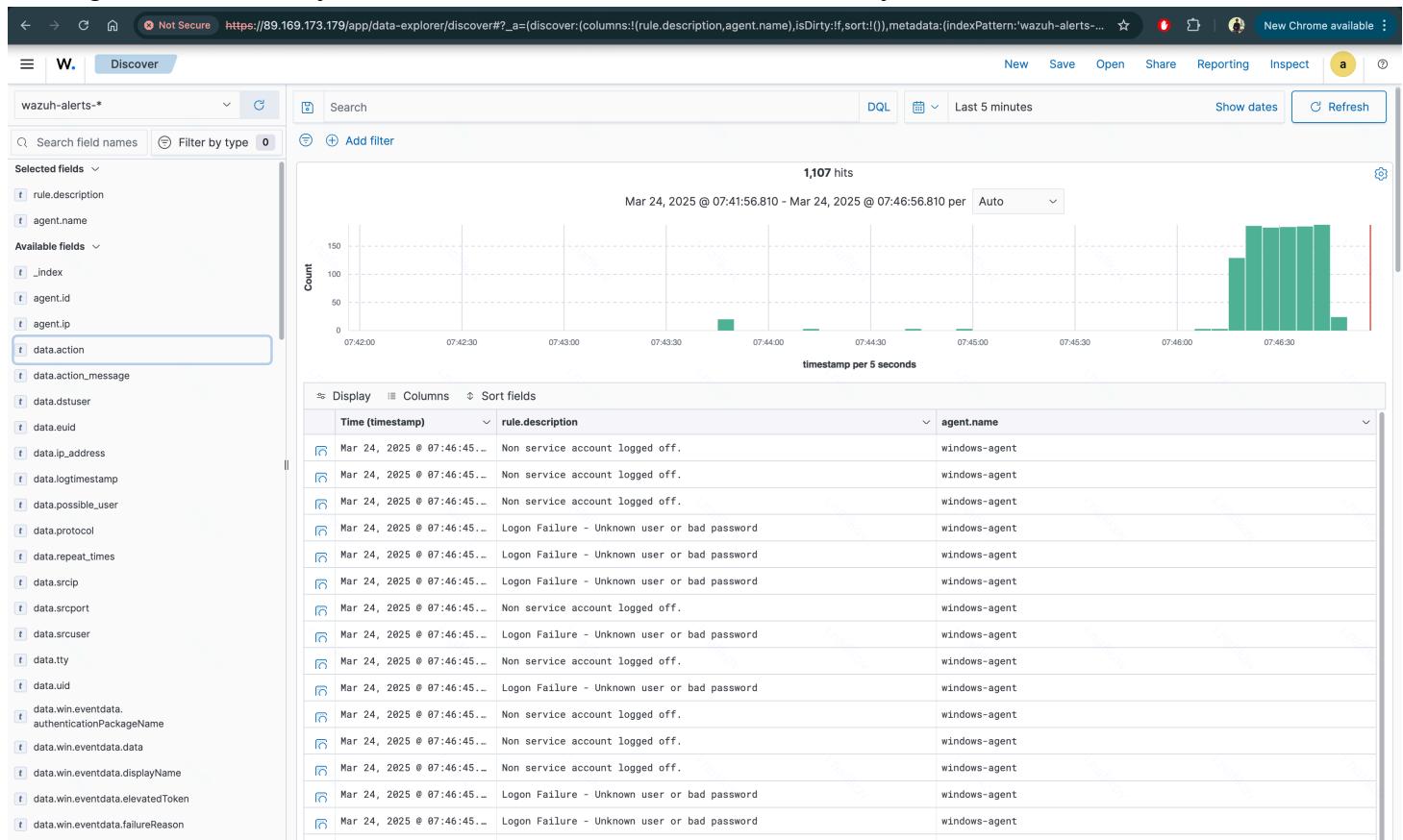


- I ran `hydra` from another laptop inside the same LAN

```
i.nafikov@macbook-KN70WXZPH lab-04 % hydra -L logins -x 2:2:1 172.20.10.7 ssh
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, this
e *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-03-24 07:46:15
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 400 login tries (l:4:p:100), ~25 tries per task
[DATA] attacking ssh://172.20.10.7:22/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-03-24 07:46:51
i.nafikov@macbook-KN70WXZPH lab-04 %
```

- And again I could easily notice the attack in the Wazuh Discovery



## Part B

### Task 5 - SOC integrations

a.

#### Task description

- a. Integrate the SIEM with a case management system of your choice? e.g theHive. Show that you are able to automatically open cases from SIEM alerts.

- I selected [TheHive](#) from StrangeBee as an Incident Response Platform and Case Management System.

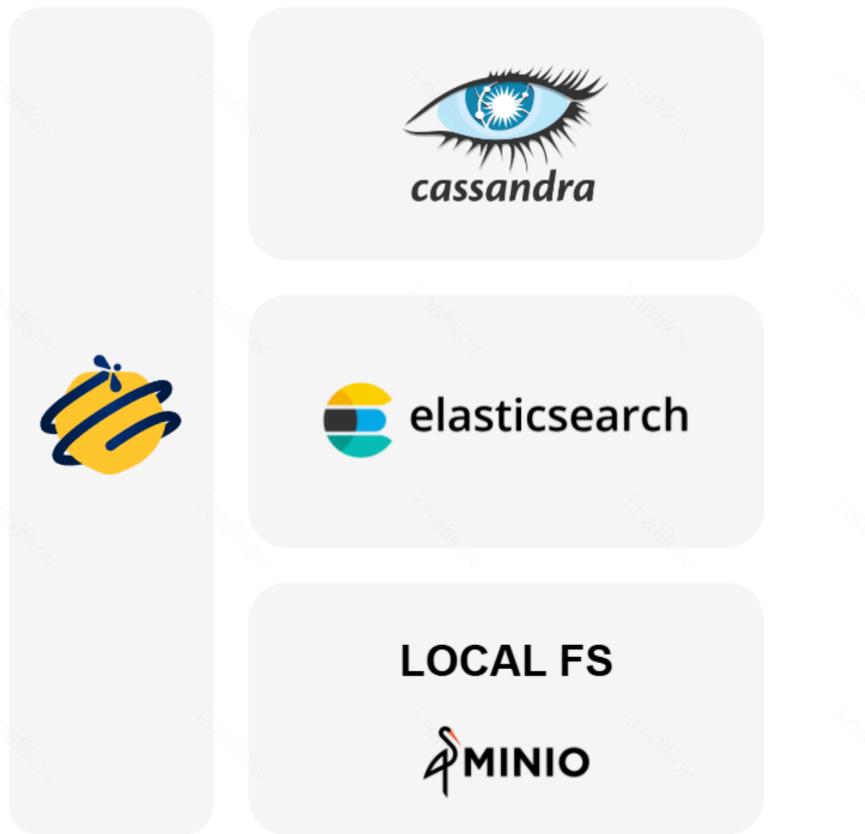
**TheHive** is an open-source incident response platform designed to assist security teams in managing and responding to security incidents effectively. It provides a collaborative environment where analysts can create, track, and investigate security cases using a structured workflow. TheHive allows for the integration of various security tools, enabling teams to enrich incident data, collaborate in real-time, and maintain documentation for audits. It is particularly suited for

organizations looking to enhance their incident response capabilities and improve overall threat management.

- TheHive architecture stack is modern and powerful

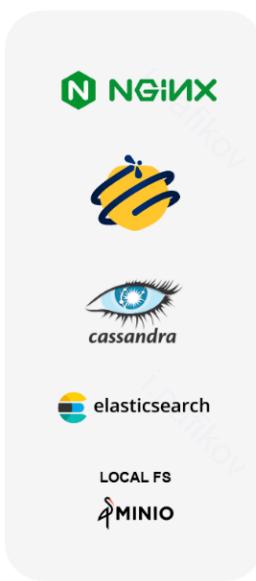
The essential components of TheHive's setup include:

-  [Apache Cassandra](#) for robust data storage, with support for version 4.x.
-  [Elasticsearch](#), serving as a powerful indexing engine, with support for versions 7.x and 8.x.
-  A file storage solution, which can be the local filesystem of the server hosting the application for standalone setups or, NFS or [S3 MINIO](#) for clustered environments.



## Deploying TheHive

- I decided to deploy TheHive as a standalone server instance which consists of the 5 main components



A standalone server setup involves installing all necessary components on a single server:

- Cassandra
- Elasticsearch
- File storage on the local filesystem (or MinIO if desired)
- TheHive
- Optional NGINX for managing HTTPS communications

For detailed installation instructions, refer to the [step-by-step installation guide](#).

- First, I deployed to a new VM on Yandex Cloud with higher resources since `theHive` is quite resource-demanding

Screenshot of the Yandex Cloud Compute Cloud Virtual machines interface. The interface shows a list of virtual machines:

Name	Cloud Backup	Status	OS	Platform	vCPU	vCPU performance	RAM	Preemptible	Disk size	Availability zone	Internal IPv4	Public IPv4	Date created	ID
thehive-server-vm		Running	Ubuntu 22.04 LTS	Intel Ice Lake	4	100%	16 GB	No	50 GB	ru-central1-b	10.129.0.31	158.160.95.136	27/03/2025, at 20:51	epd1sruijq5aluj2a4dr...
wazuh-server-ubuntu		Running	Ubuntu 22.04 LTS	Intel Ice Lake	4	100%	8 GB	No	50 GB	ru-central1-b	10.129.0.33	89.169.173.179	03/03/2025, at 21:41	epduouitjjq0m10ki3k...

- Then using the official [instruction](#) I deployed TheHive on this VM using Docker:

<https://github.com/StrangeBeeCorp/docker>

```
inafikov@compute-vm-4-16-30(ssd-1743097710847):~$ git clone https://github.com/StrangeBeeCorp/docker.git
Cloning into 'docker'...
remote: Enumerating objects: 670, done.
remote: Counting objects: 100% (44/44), done.
remote: Compressing objects: 100% (29/29), done.
remote: Total 670 (delta 18), reused 29 (delta 15), pack-reused 626 (from 1)
Receiving objects: 100% (670/670), 154.19 KiB | 1.22 MiB/s, done.
Resolving deltas: 100% (434/434), done.
inafikov@compute-vm-4-16-30(ssd-1743097710847):~$ ls docker
README.md prod1-cortex prod1-thehive prod2-cortex prod2-thehive testing versions.env
inafikov@compute-vm-4-16-30(ssd-1743097710847):~/docker$ cd testing; ls
README.md cassandra certificates cortex docker-compose.yml dot.env.template elasticsearch nginx scripts thehive
inafikov@compute-vm-4-16-30(ssd-1743097710847):~/docker/testing$ bash ./scripts/init.sh
```

```
[✓] Updated file permissions for: ./cortex/config/logback.xml
[✓] Updated file permissions for: ./cortex/config/secret.conf.template
[✓] Updated file permissions for: ./cortex/neurons/.gitkeep
[✓] Updated file permissions for: ./elasticsearch/logs/.gitkeep
[✓] Updated file permissions for: ./elasticsearch/data/.gitkeep
[✓] Updated file permissions for: ./nginx/certs/.gitkeep
[✓] Updated file permissions for: ./nginx/templates/.gitkeep
[✓] Updated file permissions for: ./nginx/templates/default.conf.template
[✓] Updated file permissions for: ./thehive/logs/.gitkeep
[✓] Updated file permissions for: ./thehive/config/application.conf
[✓] Updated file permissions for: ./thehive/config/logback.xml
[✓] Updated file permissions for: ./thehive/config/secret.conf.template
[✓] Updated file permissions for: ./thehive/data/files/.gitkeep
[✓] Updated file permissions for: ./scripts/test_init_cortex.sh
[✓] Updated file permissions for: ./scripts/output.sh
[✓] Updated file permissions for: ./scripts/init.sh
[✓] Updated file permissions for: ./scripts/test_init_thehive.sh
[✓] Updated file permissions for: ./scripts/generate_certs.sh
[✓] Updated file permissions for: ./scripts/check_permissions.sh
[✓] Updated file permissions for: ./scripts/reset.sh
[✓] Updated file permissions for: ./scripts/restore.sh
[✓] Updated file permissions for: ./scripts/backup.sh
[✓] Updated file permissions for: ./scripts/test_init_applications.sh
[✓] Permissions have been updated for files and directories.
```

[\*] Define the hostname used to connect to this server

Server Name (default: compute-vm-4-16-30-ssd-1743097710847 ): the-hive-server

[\*] No custom certificate found.  
[\*] If you want to provide your custom certificate for the Reverse Proxy, please copy the following files in the `/certificates` directory:

[\*] If you want to provide your  
\* ./certificates/server.crt  
\* ./certificates/server.key  
\* ./certificates/ca.pem

Generating self-signed certificate...

[✓] Self-signed certificate generated for compute-vm-4-16-30-ssd-1743097710847.

[✓] Initialisation completed.

[\*] Run the following command to start applications:  
\$ ./start

```
$ docker compose up
```

```
inafikov@compute-vm-4-16-30-ssd-1743097710847:~/docker/testing$
```

~ — inafikov@compute-vm-4-16-30-ssd-1743097710847: ~/docker/testing — ssh -i .ssh/theHive\_server inafikov@158.160.95.13

```
[+] Running 37/49
.. nginx [██████████] 70.12MB / 72.93MB Pulling          23.5s
.. cassandra [██████████] 130.4MB / 155.9MB Pulling      23.5s
.. thehive [██████████] 610.2MB / 616.3MB Pulling        23.5s
.. cortex [██████████] 429MB / 432.7MB Pulling           23.5s
.. elasticsearch [██████] Pulling                         23.5s
```

~ — inafikov@compute-vm-4-16-30-ssd-1743097710847: ~/docker/testing — ssh -i .ssh/theHive\_server inafikov@158.160.95.136

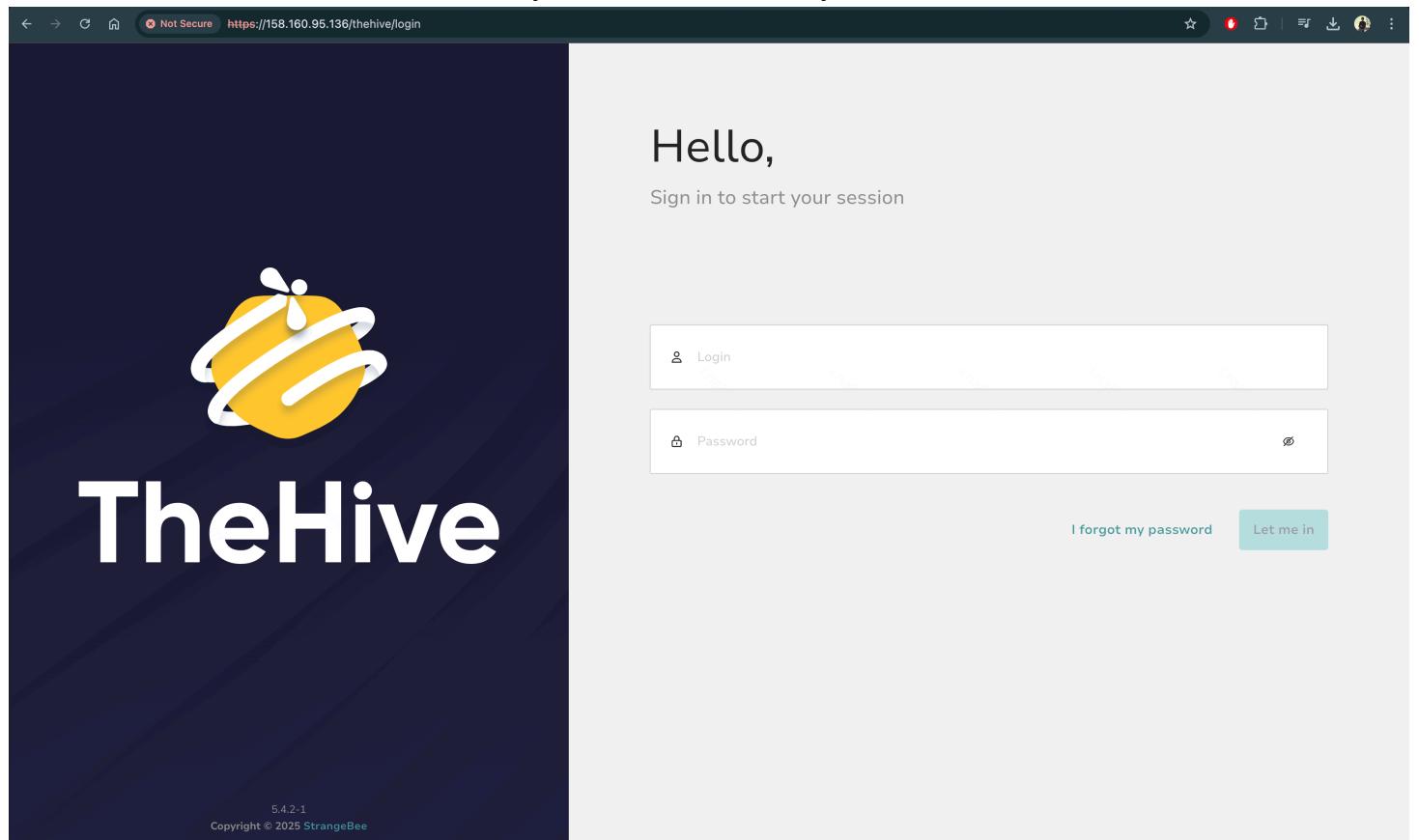
```
[+] Running 49/49
✓ nginx Pulled                                         105.1s
✓ cassandra Pulled                                      147.9s
✓ thehive Pulled                                         103.9s
✓ cortex Pulled                                         76.8s
✓ elasticsearch Pulled                                  170.9s
```

[+] Running 6/6

✓ Network testing_thehive-cortex-network	Created	7.9s
✓ Container cassandra	Healthy	127.9s
✓ Container elasticsearch	Healthy	73.9s
✓ Container cortex	Started	22.4s
✓ Container thehive	Started	76.3s
✓ Container nginx	Started	76.4s

```
inafikov@compute-vm-4-16-30-ssd-1743097710847:~/docker/testing$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS                               NAMES
cd415dabc915      nginx:1.27.2       "/docker-entrypoint..."   About a minute ago   Up 39 seconds          80/tcp, 0.0.0.0:443->443/tcp, [:]443->443/tcp   nginx
f0b38531551f      strangebee/thehive:5.4.2-1   "/opt/thehive/entryp..."   About a minute ago   Up 39 seconds (health: starting)  0.0.0.0:9000->9000/tcp   thehive
d91410648f18      thehiveproject/cortex:3.1.8-1   "/opt/cortex/entryp..."   About a minute ago   Up About a minute (healthy)    0.0.0.0:9001->9001/tcp   cortex
82bc2d561449      cassandra:4.1.7        "/docker-entrypoint.s..."   2 minutes ago       Up About a minute (healthy)    7000-7001/tcp, 7199/tcp, 9042/tcp, 9160/tcp   cassandra
80499323e7ef      elasticsearch:7.17.24     "/bin/tini -- /usr/l..."   2 minutes ago       Up About a minute (healthy)    9200/tcp, 9300/tcp   elasticsearch
inafikov@compute-vm-4-16-30-ssd-1743097710847:~/docker/testing$
```

- Afterwards, I was able to connect to my TheHive instance by the machine's IP address



## TheHive Configuration

- I logged in using default credentials as an admin

The screenshot shows the 'Organisation List' page in TheHive. The top navigation bar includes the URL 'https://158.160.95.136/thehive/administration/organisations'. A red banner at the top states, 'This instance uses a **Platinum** License for **Trial** purpose, and will expire in **15** days. [Register now.](#)' The main content area displays a table with one row of data:

	Name	Created by	Created date
<input checked="" type="checkbox"/>	admin	TheHive system user	27/03/2025 21:18

The sidebar on the left contains icons for Home, Logins, Organisations, Cases, and Dashboards. At the bottom left, it says '5.4.2-1'. At the bottom right, there are buttons for 'Previous', 'Next >', 'Show 30', and a dropdown menu.

- Here we can see the `admin` organisation with the single `admin` user

- However, I created a new separate organisation lab-org

The screenshot shows the 'Organisation List' page on the left and the 'Organisation preview' page on the right. The preview page details the newly created organization 'lab-org' with ID -4083944, created by Iskander Nafikov on 27/03/2025 22:44. The organization is active and has no linked organizations. It has two sharing rules: 'autoShare' for both 'Tasks sharing rule' and 'Observables sharing rule'. A description is provided: 'IU SD course Lab-01 Organization'. Under 'Users', one user 'Iskander Nafikov' is listed with the role 'analyst'. There are no links or alerts listed.

- And a new user inside this organisation with the role analyst which gives rights to write/read alerts inside the organisation using UI and REST API

The screenshot shows the 'Users' page for the 'lab-org' organization. A new user 'Iskander Nafikov' is being created with ID -3928160. The user is active and was created by Iskander Nafikov on 27/03/2025 22:45. The user's details include a placeholder profile picture, the name 'Iskander Nafikov', and the email 'nafikov.iskander@mail.ru'. The user is assigned the 'Normal' type and 'No' MFA. An API key is generated and can be renewed, revealed, or revoked. The user has the 'analyst' role. The permissions listed are extensive, covering various TheHive features like managing cases, alerts, and observables. A red 'Delete user' button is at the bottom.

- I created an API key to support integration of Wazuh -> TheHive since Wazuh will call TheHive's API to create an alert when Wazuh rule is triggered

Not Secure 158.160.95.136:9000/thehive/administration/organisations/lab-org/users

This instance uses a Platinum License for Trial purpose, and will expire on 27/03/2025 22:45.

**Organisation List / lab-org / Users**

**lab-org**

**Creation date**: 27/03/2025 22:44 (20 minutes ago)

**Description**: IU SD course Lab-01 Organization

**Tasks sharing rule**: autoShare

**Observables sharing rule**: autoShare

**Users** **Linked organisations**

**Details** **Full Name** **Login**

**Iskander Nafikov** nafikov.iskander@mail.ru

**Name**: Iskander Nafikov

**Created by**: Iskander Nafikov

**Created at**: 27/03/2025 22:45

**Updated at**: 27/03/2025 23:03

**Login**: nafikov.iskander@mail.ru

**Email**: nafikov.iskander@mail.ru

**Type**: Normal

**Locked**:

**MFA**: No

**API Key**: CpWx2N0RGzJ7pp040YG28Gsf8FaTp1k1

**Renew** **Reveal** **Revoke**

**Profile**: analyst

**Permissions**

```
accessTheHiveFS manageAction manageAlert/create manageAlert/delete manageAlert/import
manageAlert/reopen manageAlert/update manageAnalyse manageCase/changeOwnership manageCase/create
manageCase/delete manageCase/merge manageCase/reopen manageCase/update manageCaseReport
manageComment manageCustomEvent manageDashboard manageFunction/invoke manageKnowledgeBase
```

**Delete user**

**success** Copied to clipboard!

## Integration with Wazuh

- To integrate Wazuh with TheHive I used the following [instruction](#) from Wazuh blog written by our student and TA, **Awwal Ishiaku**

Using Wazuh and TheHive for threat protection and incident response

April 6th 2022 by Awwal Ishiaku | Wazuh 4.2



- I installed `thehive4py` Python module on my Wazuh manager

```
/var/ossec/framework/python/bin/pip3 install thehive4py==1.8.1
```

```

-- nafikov@compute-vm-4-16-30-ssd-1743097710847:~/docker/testing -- ssh -i .ssh/theHive_server nafikov@158.160.95.136
nafikov@wazuh-server-ubuntu:~$ docker exec -it single-node-wazuh_manager-1 bash
bash-5.2# /var/ossec/framework/python/bin/pip3 install thehive4py==1.8.1
Collecting thehive4py==1.8.1
  Downloading thehive4py-1.8.1-py3-none-any.whl.metadata (4.3 kB)
Requirement already satisfied: future in /var/ossec/framework/python/lib/python3.10/site-packages (from thehive4py==1.8.1) (0.18.3)
Requirement already satisfied: requests in /var/ossec/framework/python/lib/python3.10/site-packages (from thehive4py==1.8.1) (2.32.2)
Collecting python-magic (from thehive4py==1.8.1)
  Downloading python_magic-0.4.27-py2.py3-none-any.whl.metadata (5.8 kB)
Requirement already satisfied: charset-normalizer<4,>=2 in /var/ossec/framework/python/lib/python3.10/site-packages (from requests->thehive4py==1.8.1) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in /var/ossec/framework/python/lib/python3.10/site-packages (from requests->thehive4py==1.8.1) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in /var/ossec/framework/python/lib/python3.10/site-packages (from requests->thehive4py==1.8.1) (2.2.2)
Requirement already satisfied: certifi>=2017.4.17 in /var/ossec/framework/python/lib/python3.10/site-packages (from requests->thehive4py==1.8.1) (2024.7.4)
Downloaded thehive4py-1.8.1-py3-none-any.whl (32 kB)
Downloading python_magic-0.4.27-py2.py3-none-any.whl (13 kB)
Installing collected packages: python-magic, thehive4py
Successfully installed python-magic-0.4.27 thehive4py-1.8.1
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
[notice] A new release of pip is available: 23.3.2 > 25.0.1
[notice] To update, run: /var/ossec/framework/python/bin/python3.10 -m pip install --upgrade pip
bash-5.2#

```

- Then I created a Python in script `/var/ossec/integrations/custom-w2thive.py` to send alerts from Wazuh to TheHive

```

-- nafikov@compute-vm-4-16-30-ssd-1743097710847:~/docker/testing -- ssh -i .ssh/theHive_server nafikov@158.160.95.136
bash-5.2# vim /var/ossec/integrations/custom-w2thive.py
bash-5.2# head vim /var/ossec/integrations/custom-w2thive.py
head: cannot open 'vim' for reading: No such file or directory
=> /var/ossec/integrations/custom-w2thive.py <=
#!/var/ossec/framework/python/bin/python3
import json
import sys
import os
import re
import logging
import uuid
from thehive4py.api import TheHiveApi
from thehive4py.models import Alert, AlertArtifact
bash-5.2# tail /var/ossec/integrations/custom-w2thive.py
if __name__ == "__main__":
    try:
        logger.debug('debug mode') # if debug enabled
        main(sys.argv)
    except Exception:
        logger.exception('EGOR')
bash-5.2#

```

- Then I created a bash script that will run the Python script properly

```

-- nafikov@compute-vm-4-16-30-ssd-1743097710847:~/docker/testing -- ssh -i .ssh/theHive_server nafikov@158.160.95.136
# Copyright (C) 2015-2020, Wazuh Inc.
# Created by Wazuh, Inc. <info@wazuh.com>.
# This program is free software; you can redistribute it and/or modify it under the terms of GP>
WPYTHON_BIN="framework/python/bin/python3"
SCRIPT_PATH_NAME="$0"
DIR_NAME="$(dirname ${SCRIPT_PATH_NAME}); pwd -P"
bash-5.2# tail /var/ossec/integrations/custom-w2thive
  if [ -z "${WAZUH_PATH}" ]; then
    WAZUH_PATH="$(cd ${DIR_NAME}/..; pwd)"
  fi
  PYTHON_SCRIPT="${DIR_NAME}/${SCRIPT_NAME}.py"
  ;;
esac

${WAZUH_PATH}/${WPYTHON_BIN} ${PYTHON_SCRIPT} $@
bash-5.2#

```

- I gave the proper permissions to these scripts for Wazuh to be able run them

```

-- nafikov@compute-vm-4-16-30-ssd-1743097710847:~/docker/testing -- ssh -i .ssh/theHive_server nafikov@158.160.95.136
bash-5.2# chmod 755 /var/ossec/integrations/custom-w2thive.py
bash-5.2# chmod 755 /var/ossec/integrations/custom-w2thive
bash-5.2# chown root:ossec /var/ossec/integrations/custom-w2thive.py
chown: invalid group: 'root:ossec'
bash-5.2# chown root:wazuh /var/ossec/integrations/custom-w2thive.py
bash-5.2# sudo chown root:wazuh var/ossec/integrations/custom-w2thive
bash: sudo: command not found
bash-5.2# chown root:wazuh var/ossec/integrations/custom-w2thive
bash-5.2#

```

- Afterwards, to allow Wazuh to run the integration script, I added the `integration` block to the manager configuration file located at the `/var/ossec/etc/ossec.conf`. I inserted the IP address for TheHive server along with the API key that was generated earlier.
- However, since I deployed Wazuh as a Docker container I used mounted volume and to modify Wazuh Manager config

```
-- inafikov@compute-vm-4-16-30-ssd-1743097710847:~/docker/testing -- ssh -i .ssh/theHive_server inafikov@158.160.95.136 ... ov@wazuh-server-ubuntu: ~/wazuh-docker/single-node/config/wazuh_cluster -- ssh -i .ssh/wazuh_server nafikov@89.169.173.179
nafikov@wazuh-server-ubuntu:~/wazuh-docker/single-node/config/wazuh_cluster$ vim wazuh_manager.conf
nafikov@wazuh-server-ubuntu:~/wazuh-docker/single-node/config/wazuh_cluster$ cat wazuh_manager.conf | grep "<integration>" -A 7
<integration>
  <name>custom-w2thive</name>
  <hook_url>http://158.160.95.136:9000/thehive</hook_url>
  <api_key>cpxX2N0RGzJ7pp040YG28Gsf8FaTlpk1</api_key>
  <alert_format>json</alert_format>
</integration>

</ossec_config>
nafikov@wazuh-server-ubuntu:~/wazuh-docker/single-node/config/wazuh_cluster$
```

- And restarted Wazuh Manager

```
-- inafikov@compute-vm-4-16-30-ssd-1743097710847:~/docker/testing -- ssh -i .ssh/theHive_server inafikov@158.160.95.136 ... ov@wazuh-server-ubuntu: ~/wazuh-docker/single-node/config/wazuh_cluster -- ssh -i .ssh/wazuh_server nafikov@89.169.173.179
nafikov@wazuh-server-ubuntu:~/wazuh-docker/single-node/config/wazuh_cluster$ docker compose restart wazuh.manager
WARN[0000] /home/nafikov/wazuh-docker/single-node/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Restarting 2/2
  ✓ Container single-node-wazuh.manager-1    Started
  ✓ Container single-node-wazuh.dashboard-1  Started
nafikov@wazuh-server-ubuntu:~/wazuh-docker/single-node/config/wazuh_cluster$
```

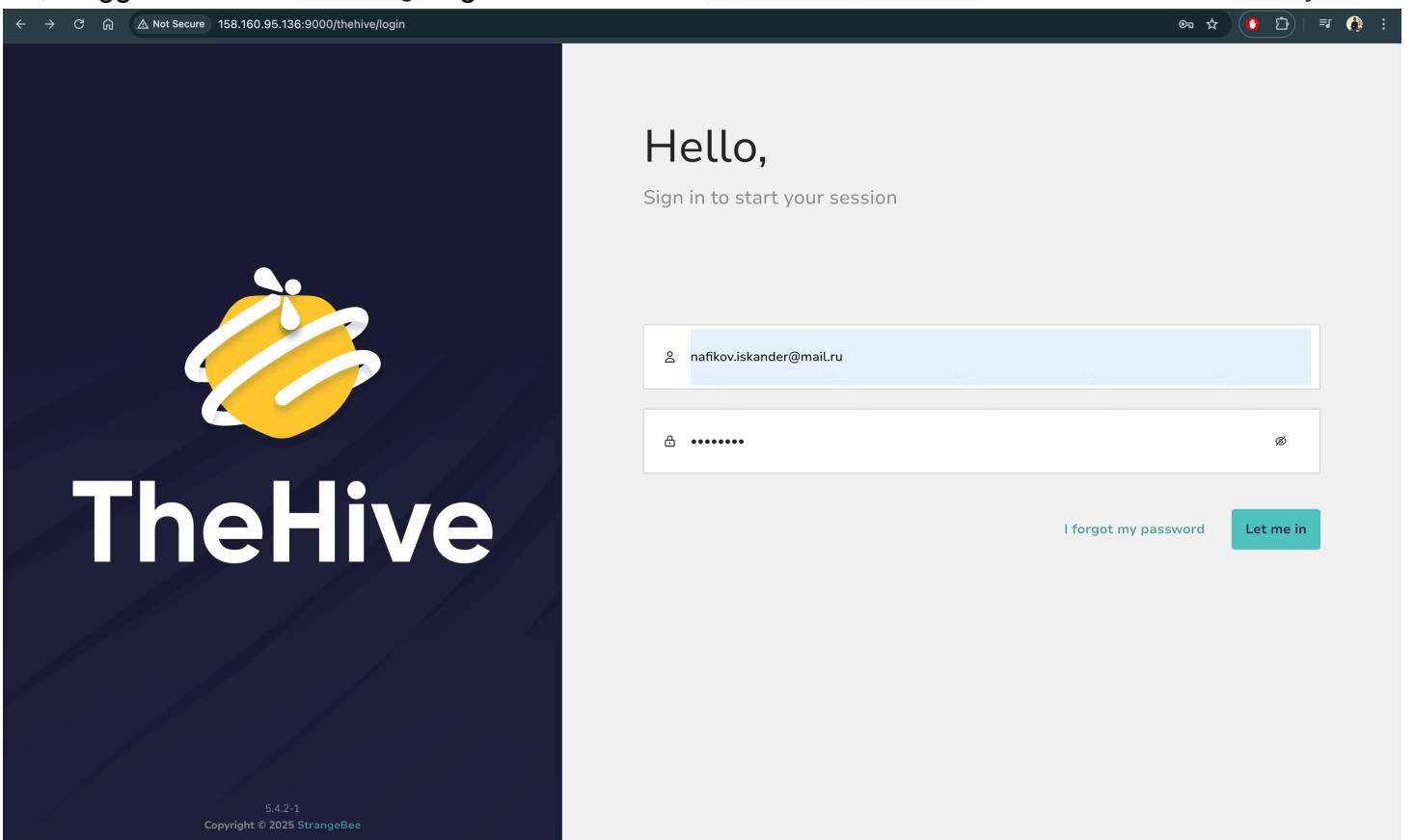
- Finally, the file changes were applied to the `/var/ossec/etc/ossec.conf` inside the container

```
-- inafikov@compute-vm-4-16-30-ssd-1743097710847:~/docker/testing -- ssh -i .ssh/theHive_server inafikov@158.160.95.136 ... ov@wazuh-server-ubuntu: ~/wazuh-docker/single-node/config/wazuh_cluster -- ssh -i .ssh/wazuh_server nafikov@89.169.173.179
nafikov@wazuh-server-ubuntu:~/wazuh-docker/single-node/config/wazuh_cluster$ docker exec single-node-wazuh.manager-1 cat /var/ossec/etc/ossec.conf | grep "<integration>" -A 7
<integration>
  <name>custom-w2thive</name>
  <hook_url>http://158.160.95.136:9000/thehive</hook_url>
  <api_key>cpxX2N0RGzJ7pp040YG28Gsf8FaTlpk1</api_key>
  <alert_format>json</alert_format>
</integration>

</ossec_config>
nafikov@wazuh-server-ubuntu:~/wazuh-docker/single-node/config/wazuh_cluster$
```

## Alerts analysis & Incident Response

- So, I logged in to the `lab-org` organisation as the `Iskander Nafikov` user with the role `analyst`



- Finally, I was able to check the alerts inside TheHive coming from Wazuh

This screenshot shows a list of alerts in TheHive interface. The alerts are from the 'wazuh\_alert' type, generated by the 'wazuh' source. Each alert has a timestamp of 'New a few seconds ago'. The alerts are triggered by rule 598, which monitors registry key entries. The details column shows the rule ID, agent IP (10.91.53.142), agent ID (001), and agent name (windows-agent). The observables and TTPs columns are both empty. The assignee, dates, and other details are also listed.

Status	Severity	Title	# Case	Type	Source	Reference	Details	Assignee	Dates
New	M	Registry Key Entry Added to the System	538031	wazuh_alert	wazuh		Observables: 1 TTPs: 0	O. 27/03/2025 23:21 C. 27/03/2025 23:21 U. 27/03/2025 23:21	
New	M	Registry Key Entry Added to the System	021bb4	wazuh_alert	wazuh		Observables: 1 TTPs: 0	O. 27/03/2025 23:21 C. 27/03/2025 23:21 U. 27/03/2025 23:21	
New	M	Registry Key Integrity Checksum Changed	4aff3c	wazuh_alert	wazuh		Observables: 1 TTPs: 0	O. 27/03/2025 23:21 C. 27/03/2025 23:21 U. 27/03/2025 23:21	
New	M	Registry Key Entry Added to the System	65a561	wazuh_alert	wazuh		Observables: 1 TTPs: 0	O. 27/03/2025 23:21 C. 27/03/2025 23:21 U. 27/03/2025 23:21	
New	M	Registry Key Integrity Checksum Changed	0683e5	wazuh_alert	wazuh		Observables: 1 TTPs: 0	O. 27/03/2025 23:21 C. 27/03/2025 23:21 U. 27/03/2025 23:21	
New	M	Registry Key Integrity Checksum Changed	d9b6a9	wazuh_alert	wazuh		Observables: 1 TTPs: 0	O. 27/03/2025 23:21 C. 27/03/2025 23:21 U. 27/03/2025 23:21	
New	M	Registry Key Integrity Checksum Changed	bc101a	wazuh_alert	wazuh		Observables: 1 TTPs: 0	O. 27/03/2025 23:21 C. 27/03/2025 23:21 U. 27/03/2025 23:21	

- We see that these alerts came right from Wazuh in the same format

This screenshot shows the details of a specific alert (#538031) in TheHive. The alert is titled 'Registry Key Entry Added to the System'. The general tab shows the alert ID (10076344), created by 'Iskander Nafikov' at '27/03/2025 23:21'. The observables and TTPs sections are empty. The description section contains a timestamp table. The rule section lists various parameters like rule.level (5), rule.description (Registry Key Entry Added to the System), and rule.id (598). The comments section is empty.

key	val
timestamp	2025-03-27T20:10:53.658+0000

key	val
rule.level	5
rule.description	Registry Key Entry Added to the System
rule.id	598
rule.mitre.id	['T1112']
rule.mitre.tactic	['Defense Evasion']
rule.mitre.technique	['Modify Registry']
rule.firetimes	52
rule.mail	False
rule.groups	['ossec', 'syscheck', 'syscheck_entry_added', 'syscheck_registry']
rule_pci_dss	['11.5']
rule_cvn13	['A 1 2']

- I even created a new case with low severity from some alert

This instance uses a **Platinum License** for **Trial** purpose, and will expire in **15 days**. [Register now.](#)

**Cases / #1 / Description**

**#1 Windows: Service startup type was changed**

**General** Tasks (0) Observables (0) TTPs (0) Attachments Timeline Report Pages History Linked alerts

**Title**  
Windows: Service startup type was changed

**Tags**  
rule=61104 | agent\_ip=10.91.53.142 | agent\_id=001 | agent\_name=wazuh

**Description**

**Timestamp**

key	val
timestamp	2025-03-27T20:28:30.893+0000

**Rule**

key	val
rule.level	3
rule.description	Service startup type was changed
rule.id	61104
rule.info	This does not appear to be logged on Windows 2000
rule.firetimes	3
rule.mail	False
rule.groups	['windows', 'windows_system', 'policy_changed']
rule.pci_dss	['10.6']
rule.gdpr	['IV_35.7.d']

**Comments**

The incident started  
27/03/2025 23:38

Getting to investigation  
27/03/2025 23:38

Type a comment...  
Hit "SHIFT + ENTER" for a new line

- Then I closed it quickly emulating the case was false positive

This instance uses a **Platinum License** for **Trial** purpose, and will expire in **15 days**. [Register now.](#)

**Cases / #1 / Description**

**#1 Windows: Service startup type was changed**

**General** Tasks (0) Observables (0) TTPs (0) Attachments Timeline Report Pages History Linked alerts

**data.win.system.processID** 904

**data.win.system.threadID** 4648

**data.win.system.channel** System

**data.win.system.computer** DESKTOP-KB1GRFV

**data.win.system.severityValue** INFORMATION

**data.win.system.message** "The start type of the Background Intelligent Transfer Service service was changed from auto start to demand start."

**data.win.eventdata.param1** Background Intelligent Transfer Service

**data.win.eventdata.param2** auto start

**data.win.eventdata.param3** demand start

**data.win.eventdata.param4** BITS

**Location**

key	val
location	EventChannel

**Summary**

It was okay, just false positive alert triggering. I will change the rule trigger in Wazuh

**Comments**

The incident started  
27/03/2025 23:38

Getting to investigation  
27/03/2025 23:38

It's all okay, closing the case  
27/03/2025 23:39

Type a comment...  
Hit "SHIFT + ENTER" for a new line