

LAB4 SIEM TERUNGWA

AKOR JACOB

🕒 Created	@February 27, 2025 3:42 PM
☑ Attendance Required	<input type="checkbox"/>

Part A

Task 1 - Introduction

a. Brief Explanation of the Architecture of my SIEM Solution

- **Architecture of Wazuh SIEM**

Wazuh is an open-source security platform that provides threat detection, visibility, and compliance monitoring

The architecture of Wazuh consists of the following components:

1. **Wazuh Manager:** The central component that collects, analyzes, and stores data from agents. It also performs correlation and rule-based alerting and integrates with external tools like VirusTotal, YARA, and osquery.
2. **Wazuh Agents:** Lightweight software installed on endpoints (e.g., Windows, Linux, network devices) that collect and forward logs, file integrity data, and system inventory to the Wazuh Manager.
3. **Elastic Stack (Elasticsearch, Logstash, Kibana):** Wazuh integrates with the Elastic Stack for data storage, visualization, and advanced analytics. Elasticsearch stores the data, Logstash processes it, and Kibana provides a user-friendly dashboard for visualization.
4. **External Integrations:** Wazuh can integrate with tools like VirusTotal for malware analysis, YARA for threat detection, and osquery for endpoint monitoring.

b. Advantages of Open Source Solutions and How Vendors Make Money

- **Cost-Effective:** No licensing fees, reducing total cost of ownership.
- **Transparency & Security:** Open-source code allows for auditing and verifying security features.
- **Community Support & Customizability:** Large developer communities provide updates, plugins, and integrations.

How Vendors Make Money:

- **Support and Consulting:** Vendors offer paid support, consulting, and training services.
- **Enterprise Features:** Some open-source solutions offer premium features or enterprise versions with additional capabilities.
- **Managed Services:** Vendors may offer managed SIEM services where they handle the deployment, monitoring, and maintenance for clients.

Task 2 - Setup infrastructure

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

- For a Linux Machine, I installed the Wazuh Manager, Wazuh Indexer (Elasticsearch), and Wazuh Dashboard (Kibana).
 - `curl -sO https://packages.wazuh.com/4.7/wazuh-install.sh && sudo bash ./wazuh-install.sh --all-in-one`

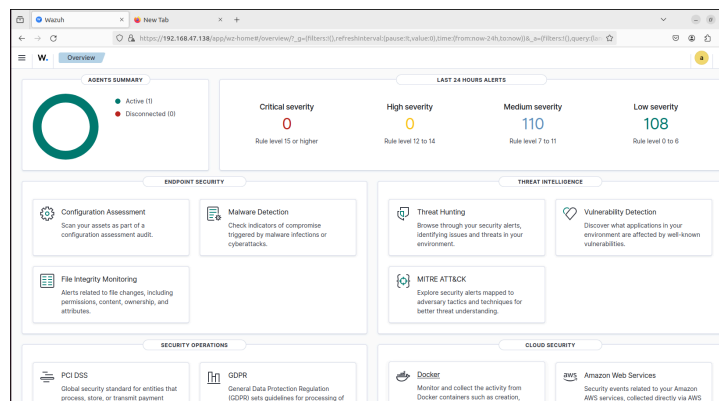
```

jacob@jacob-virtual-machine: ~
jacob@jacob-virtual-machine:~$ curl -sO https://packages.wazuh.com/4.7/wazuh-install.sh && sudo bash ./wazuh-install.sh --all-in-one
27/02/2025 18:18:41 INFO: Starting Wazuh installation assistant. Wazuh version: 4.7.5
27/02/2025 18:18:44 INFO: Verbose logging redirected to /var/log/wazuh-install.log
27/02/2025 18:18:48 INFO: --- Dependencies ---
27/02/2025 18:18:48 INFO: Installing gawk.
27/02/2025 18:18:55 INFO: Wazuh web interface port will be 443.
27/02/2025 18:19:00 INFO: --- Dependencies ---
27/02/2025 18:19:00 INFO: Installing apt-transport-https.
27/02/2025 18:19:10 INFO: Wazuh repository added.
27/02/2025 18:19:10 INFO: --- Configuration files ---
27/02/2025 18:19:10 INFO: Generating configuration files.
27/02/2025 18:19:12 INFO: Created wazuh-install-files.tar. It contains the Wazuh cluster key, certificates, and passwords necessary for installation.
27/02/2025 18:19:13 INFO: --- Wazuh indexer ---
27/02/2025 18:19:13 INFO: Starting Wazuh Indexer installation.
27/02/2025 18:21:16 INFO: Wazuh Indexer installation finished.
27/02/2025 18:21:16 INFO: Wazuh Indexer post-install configuration finished.
27/02/2025 18:21:16 INFO: Starting service wazuh-indexer.
27/02/2025 18:22:15 INFO: wazuh-indexer service started.
27/02/2025 18:22:15 INFO: Initializing Wazuh Indexer cluster security settings.
27/02/2025 18:22:27 INFO: Wazuh Indexer cluster initialized.
27/02/2025 18:22:27 INFO: --- Wazuh server ---
27/02/2025 18:22:27 INFO: Starting the Wazuh manager installation.
27/02/2025 18:24:37 INFO: Wazuh manager installation finished.
27/02/2025 18:24:37 INFO: Starting service wazuh-manager.
27/02/2025 18:24:57 INFO: wazuh-manager service started.
27/02/2025 18:24:57 INFO: Starting filebeat installation.
27/02/2025 18:25:14 INFO: Filebeat installation finished.
27/02/2025 18:25:16 INFO: Filebeat post-install configuration finished.
27/02/2025 18:25:16 INFO: Starting service filebeat.
27/02/2025 18:25:18 INFO: filebeat service started.
27/02/2025 18:25:18 INFO: --- Wazuh dashboard ---
27/02/2025 18:25:18 INFO: Starting Wazuh dashboard installation.
27/02/2025 18:26:56 INFO: Wazuh dashboard installation finished.
27/02/2025 18:26:56 INFO: Wazuh dashboard post-install configuration finished.
27/02/2025 18:26:56 INFO: Starting service wazuh-dashboard.
27/02/2025 18:26:57 INFO: wazuh-dashboard service started.
27/02/2025 18:27:33 INFO: Initializing Wazuh dashboard web application.
27/02/2025 18:27:34 INFO: Wazuh dashboard web application installed.
27/02/2025 18:27:34 INFO: --- Summary ---
27/02/2025 18:27:34 INFO: You can access the web interface https://wazuh-dashboard-ip:443
User: admin
Password: k99s9MNC58x8i81+7sW0u7zk738
27/02/2025 18:27:34 INFO: --- Dependencies ---
27/02/2025 18:27:34 INFO: Removing gawk.
27/02/2025 18:27:44 INFO: Installation finished.
jacob@jacob-virtual-machine:~$

```

I accessed the Wazuh Dashboard by navigating to a browser using wazuh-dashboard ip address and the port number 443

- <https://192.168.47.138:443>



I verified if the Wazuh Manager is running by checking the status:

- `sudo systemctl status wazuh-manager`

```

jacob@jacob-virtual-machine:~$ sudo systemctl status wazuh-manager
[sudo] password for jacob:
● wazuh-manager.service - Wazuh manager
   Loaded: loaded (/lib/systemd/system/wazuh-manager.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2025-02-27 18:24:57 CET; 26min ago
     Tasks: 121 (limit: 10062)
    Memory: 396.8M
      CPU: 1min 40.500s
   CGroup: /system.slice/wazuh-manager.service
           └─50294 /var/ossec/framework/python/bin/python3 /var/ossec/apl/scripts/wazuh-aptd.py
             └─50334 /var/ossec/bin/wazuh-authd
               └─50349 /var/ossec/bin/wazuh-db
                 └─50374 /var/ossec/bin/wazuh-execd
                   └─50384 /var/ossec/framework/python/bin/python3 /var/ossec/apl/scripts/wazuh-aptd.py
                     └─50389 /var/ossec/framework/python/bin/python3 /var/ossec/apl/scripts/wazuh-aptd.py
                       └─50394 /var/ossec/bin/wazuh-analysisd
                         └─50396 /var/ossec/framework/python/bin/python3 /var/ossec/apl/scripts/wazuh-aptd.py
                           └─50441 /var/ossec/bin/wazuh-syscheckd
                             └─50456 /var/ossec/bin/wazuh-remoted
                               └─50488 /var/ossec/bin/wazuh-logcollector
                                 └─50508 /var/ossec/bin/wazuh-monitord
                                   └─50529 /var/ossec/bin/wazuh-modulesd

Feb 27 18:24:48 jacob-virtual-machine env[50237]: Started wazuh-db...
Feb 27 18:24:50 jacob-virtual-machine env[50237]: Started wazuh-execd...
Feb 27 18:24:51 jacob-virtual-machine env[50237]: Started wazuh-analysisd...
Feb 27 18:24:52 jacob-virtual-machine env[50237]: Started wazuh-syscheckd...
Feb 27 18:24:53 jacob-virtual-machine env[50237]: Started wazuh-remoted...
Feb 27 18:24:54 jacob-virtual-machine env[50237]: Started wazuh-logcollector...
Feb 27 18:24:55 jacob-virtual-machine env[50237]: Started wazuh-monitord...
Feb 27 18:24:55 jacob-virtual-machine env[50237]: Started wazuh-modulesd...
Feb 27 18:24:57 jacob-virtual-machine env[50237]: Completed.
Feb 27 18:24:57 jacob-virtual-machine systemd[1]: Started Wazuh manager.
jacob@jacob-virtual-machine:~$

```

- Set Up the Wazuh Agent on a Windows Machine

I downloaded and installed the Wazuh Agent by running the installer.

On the Wazuh Manager (Linux) - I generated the Agent Key:

- `sudo /var/ossec/bin/manage_agents`

```
Available agents:
ID: 001, Name: wlnserver01, IP: 192.168.47.138
Provide the ID of the agent to extract the key (or 'q' to quit): 001

Agent key information for '001' is:
MDAxIHdpbnNlcjZlcjAxIDE5Mi4xNjguNDcuMTM4IDk4N2YwZjAzZTBkNzNiZjFhY2QyYjdKM2I4ZTU5NzFjMzhkMWI3Nz
```

On the Wazuh Agent (Windows) - I registered the Agent with the Wazuh Manager using a command prompt as administrator.

- "C:\Program Files (x86)\ossec-agent\agent-auth.exe" -m 192.168.47.138 -p 1515 -A WinServer01 -k "MDAxIHdpbnNlcjZlcjAxIDE5Mi4xNjguNDcuMTM4IDk4N2YwZjAzZTBkNzNiZjFhY2QyYjdKM2I4ZTU5NzFjMzhkMWI3Nz"

Then I restarted the agent service:

- `net start WazuhSvc`
- `sc query WazuhSvc`

```
Select Administrator: Command Prompt

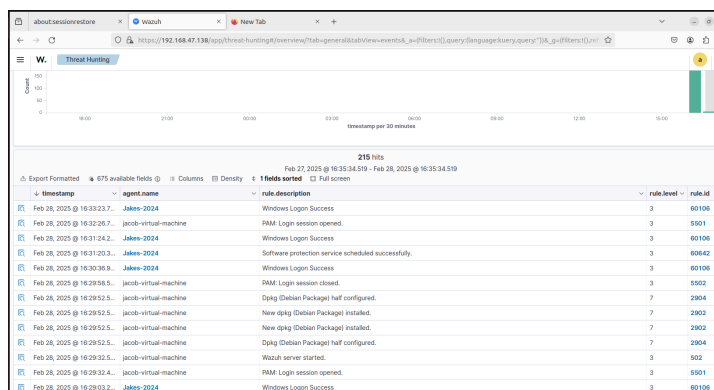
C:\Windows\System32>net start WazuhSvc
The Wazuh service is starting.
The Wazuh service was started successfully.

C:\Windows\System32>sc query WazuhSvc

SERVICE_NAME: WazuhSvc
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE                : 4   RUNNING
                        (STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE      : 0   (0x0)
        SERVICE_EXIT_CODE  : 0   (0x0)
        CHECKPOINT          : 0x0
        WAIT_HINT           : 0x0
```

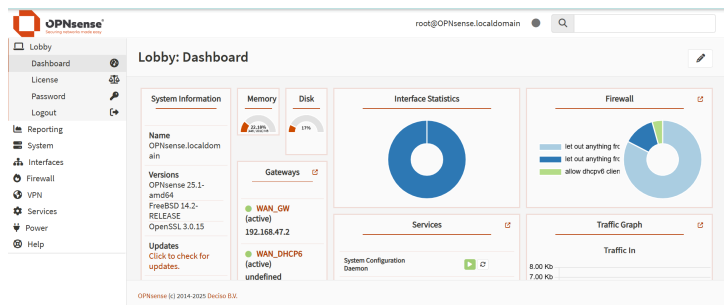
Verification of Log Collection:

- On the Wazuh Dashboard, I navigated to the **Agents** section and verified that the Windows agent is connected Then i checked for Windows Event Logs



Configuration of Network Device

- I configured OPNsense to forward its logs to the Wazuh manager by using the Wazuh agent. Below is the OPNsense Dashboard:



- After the configuration, Logs from OPNsense appeared in the Wazuh dashboard under the **Events** section, as seen below:

The screenshot shows the Wazuh Threat Hunting dashboard. The table displays a list of events with columns: timestamp, agent name, rule description, rule level, and rule id. The events are filtered by 'rule level' and 'rule id'.

timestamp	agent name	rule description	rule level	rule id
Mar 1, 2025 @ 17:12:08.960	jacob-virtual-machine	Apparmor DENIED	3	53002
Mar 1, 2025 @ 17:11:33.890	jacob-virtual-machine	Apparmor DENIED	3	53002
Mar 1, 2025 @ 17:11:15.866	jacob-virtual-machine	Possible kernel level rootkit	11	521
Mar 1, 2025 @ 17:11:09.862	jacob-virtual-machine	Apparmor DENIED	3	53002
Mar 1, 2025 @ 17:11:05.865	jacob-virtual-machine	Apparmor DENIED	3	53002
Mar 1, 2025 @ 17:11:05.865	jacob-virtual-machine	Apparmor DENIED	3	53002
Mar 1, 2025 @ 17:10:37.570	jacob-virtual-machine	PAM Login session opened.	3	5001
Mar 1, 2025 @ 17:09:47.500	jacob-virtual-machine	Wazuh server started.	3	502
Mar 1, 2025 @ 01:12:17.792	jacob-virtual-machine	pfSense firewall rules grouped.	4	87700
Mar 1, 2025 @ 01:12:17.792	jacob-virtual-machine	pfSense firewall rules grouped.	4	87700
Mar 1, 2025 @ 01:12:17.791	jacob-virtual-machine	pfSense firewall rules grouped.	4	87700
Mar 1, 2025 @ 01:12:17.791	jacob-virtual-machine	pfSense firewall rules grouped.	4	87700
Mar 1, 2025 @ 01:12:17.789	jacob-virtual-machine	pfSense firewall rules grouped.	4	87700
Mar 1, 2025 @ 01:12:17.789	jacob-virtual-machine	pfSense firewall rules grouped.	4	87700
Mar 1, 2025 @ 01:12:17.789	jacob-virtual-machine	pfSense firewall rules grouped.	4	87700
Mar 1, 2025 @ 01:12:17.789	jacob-virtual-machine	pfSense firewall rules grouped.	4	87700
Mar 1, 2025 @ 01:12:16.964	jacob-virtual-machine	pfSense firewall rules grouped.	4	87700

I was able to view one of the pfSense firewall logs:

The screenshot shows the Wazuh Document Details page. The table displays a single row of data for a pfSense firewall log. The fields include _index, _type, _id, _score, _source, and _type.

Field	Value
_index	wazuh-alerts-4.x-2025.03.01
_type	log
_id	1740787937.857907
_score	1
_source	{ "timestamp": "Mar 1 00:10:53", "agent": "OPNsense.localdomain", "filterlog": "filterlog", "data": { "action": "pass", "dstip": "192.168.47.139", "dstport": "53", "id": "0aec9733af953ca831fab85fec6280a0", "length": "36", "protocol": "udp", "srcip": "192.168.47.139", "srcport": "5800" } }
_type	log

2b. Why specifically are you able to view these logs i.e select two visible logs, explain these logs, and explain why and how you can view them on the SIEM.

Using 2 examples of the visible logs on my wazuh-dashboard

Example 1:

Mar 1, 2025 @ 01:12:17.791	jacob-virtual-machine	pfSense firewall rules grouped.	4	87700
Mar 1, 2025 @ 01:12:17.789	jacob-virtual-machine	pfSense firewall rules grouped.	4	87700
Mar 1, 2025 @ 01:12:17.789	jacob-virtual-machine	pfSense firewall rules grouped.	4	87700
Mar 1, 2025 @ 01:12:17.789	jacob-virtual-machine	pfSense firewall rules grouped.	4	87700

Explanation of pfSense Firewall Rules Grouped Logs

- **PfSense:** pfSense is an open-source firewall and router software. It logs events related to firewall rules, such as allowed or blocked traffic.
- **Firewall Rules Grouped Logs:** These logs represent events where traffic matches a specific firewall rule or group of rules. For example:
 - Traffic allowed by a rule.
 - Traffic is blocked by a rule.
 - Logs grouped by rule ID or rule description.

Why You Can View These Logs:

You can view the logs by the following reasons:

- **Centralized Log Collection:** A SIEM like Wazuh is designed to collect logs from multiple sources, including firewalls like pfSense. This centralized approach allows you to view and analyze logs from all connected devices in one place.
- **Real-Time Monitoring:** The SIEM continuously monitors logs in real time, enabling you to detect and respond to security events as they happen.
- **Normalization and Correlation:** The SIEM normalizes logs (converts them into a standard format) and correlates events across devices, making it easier to identify patterns and anomalies.

How You Can View These Logs

- **Syslog:** pfSense sends logs to the Wazuh Manager via Syslog.
- **Log Forwarding:** Logs are forwarded to the Wazuh Manager, where they are processed, analyzed, and stored.
- **Visualization:** The Wazuh Dashboard (Kibana) provides a user-friendly interface to view and analyze these logs.

Example 2: Dpkg (Debian Package) Half-Configured

Feb 28, 2025 @ 17:00:31.4...	Jakes-2024	Windows Login Success	3	60198
Feb 28, 2025 @ 17:00:22.5...	jacob-virtual-machine	Dpkg (Debian Package) half configured	7	2804

- **What It Is:** A Dpkg half-configured log indicates that a Debian package installation or update was interrupted or failed, leaving the package in a partially configured state. This log helps identify issues with software installations, which could lead to system instability or security vulnerabilities.
- **How It's Collected:**
 - The Wazuh agent on the Linux machine monitors the system logs (e.g., `/var/log/dpkg.log`).

Configuration of Linux Device

On your Fedora device, i added the Wazuh repository and installed the wazuh-agent:

- `sudo bash -c 'cat > /etc/yum.repos.d/wazuh.repo << EOF`
[wazuh]

- `sudo dnf install wazuh-agent`

I configured the Wazuh Agent by editing the Wazuh Agent configuration file (`/var/ossec/etc/ossec.conf`) to point to the Wazuh Manager.

Timeline per 30 minutes

2,603 hits

Feb 26, 2025 @ 21:43:04.334 - Mar 1, 2025 @ 21:43:04.334

Export Formatted 734 available fields Columns Density 1 fields sorted Full screen

timestamp	agent_name	rule.description	rule.level	rule.id
Mar 1, 2025 @ 21:41:32.573	jacob-virtual-machine	PAM Login session closed.	3	5902
Mar 1, 2025 @ 21:41:32.573	jacob-virtual-machine	Successful sudo to ROOT executed.	3	5402
Mar 1, 2025 @ 21:41:32.573	jacob-virtual-machine	PAM Login session opened.	3	5901
Mar 1, 2025 @ 21:26:49.615	fedora	Wazuh agent started.	3	903
Mar 1, 2025 @ 21:26:48.275	fedora	Wazuh agent stopped.	3	906
Mar 1, 2025 @ 21:25:46.331	fedora	PAM Login session closed.	3	5902
Mar 1, 2025 @ 21:25:46.009	fedora	PAM Login session opened.	3	5901
Mar 1, 2025 @ 21:25:46.056	fedora	First time user executed sudo.	4	5403
Mar 1, 2025 @ 21:25:46.041	fedora	PAM Login session closed.	3	5902
Mar 1, 2025 @ 21:25:46.037	fedora	First time user executed sudo.	4	5403
Mar 1, 2025 @ 21:25:42.981	fedora	New wazuh agent connected.	3	901

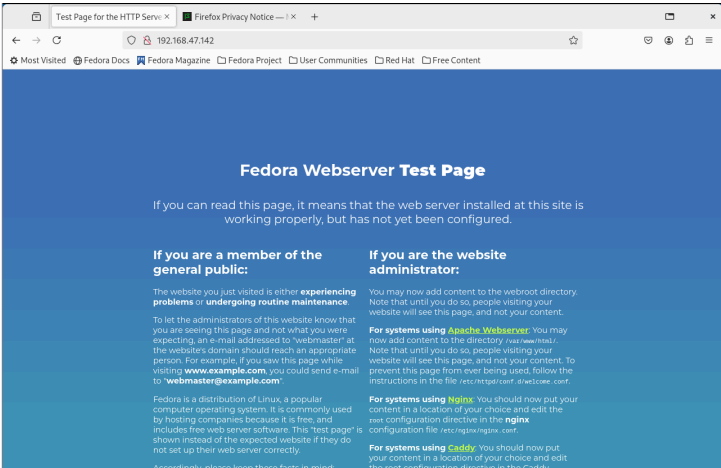
LAB4 SIEM AKOR JACOB TERUNGWA

Document Details		View surrounding documents	View single document
Table	JSON		
index	wazuh-alerts-4.x-2025.03.01		
agent.id	882		
agent.name	fedora		
data.dstuser	root		
decoder.name	pam		
decoder.parent	pam		
full_log	Mar 01 20:25:29 fedora sudo[5479]: pam_unix(sudo:session): session closed for user root		
id	1748868746.5207279		
input.type	log		
location	journald		
manager.name	jacob-virtual-machine		
predecoder.hostname	fedora		
predecoder.program_name	sudo		
predecoder.timestamp	Mar 01 20:25:29		
rule.description	PAM: Login session closed.		
rule.firetimes	2		
rule.gdpr	IV_32.2		
rule.gpg13	7.8, 7.9		
rule.groups	pam, syslog		

Task 3 - Use cases <select any 2>

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.LAB4 SIEM

- I installed and configured an Apache web server on fedora using the command:
 - sudo dnf install httpd -y
 - sudo systemctl enable --now httpd
- In the browser, I viewed the Apache landing page and verified the installation: http://192.168.47.142



I added the following to /var/ossec/etc/ossec.conf file to configure the Wazuh agent and monitor the Apache access logs:

```
<localfile>
  <log_format>syslog</log_format>
  <location>/var/log/httpd/access_logs</location>
</localfile>
```

I restarted the Wazuh agent to apply the changes:

- `sudo systemctl restart wazuh-agent`

On the Wazuh server, I added the IP address of the RHEL endpoint to a CDB list and then configured rules and Active Response using the command:

I downloaded the Alienvault IP reputation database:

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

```
jacob@jacob-virtual-machine:~$ sudo wget https://raw.githubusercontent.com/firehol/blocklist-ipsets/master/alienvault_reputation.ipset -O /var/ossec/etc/lists/alienvault_reputation.ipset
--2025-03-02 22:43:22-- https://raw.githubusercontent.com/firehol/blocklist-ipsets/master/alienvault_reputation.ipset
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.109.133, 185.199.111.133, 185.199.108.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.109.133|: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_re 100%[=====] 9,27K --KB/s in 0,003s
2025-03-02 22:43:22 (3,27 MB/s) - '/var/ossec/etc/lists/alienvault_reputation.ipset' saved [9495/9495]
```

I appended the IP address of the attacker endpoint to the IP reputation database using the command below:

- `sudo echo "192.168.47.139" >> /var/ossec/etc/lists/alienvault_reputation.ipset`

Downloaded a script to convert from the .ipset format to the .cdb list format:

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

```
jacob@jacob-virtual-machine:~$ sudo wget https://wazuh.com/resources/iplist-to-cdblist.py -O /tmp/iplist-to-cdblist.py
[sudo] password for jacob:
--2025-03-03 00:54:13-- https://wazuh.com/resources/iplist-to-cdblist.py
Resolving wazuh.com (wazuh.com)... 108.156.66.184, 108.156.66.88, 108.156.66.30, ...
Connecting to wazuh.com (wazuh.com)|108.156.66.184|: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
2025-03-03 00:54:13 (60,8 MB/s) - '/tmp/iplist-to-cdblist.py' saved [1570/1570]
jacob@jacob-virtual-machine:~$
```

I converted the alienvault_reputation.ipset file to a .cdb format using the previously downloaded script:

- `sudo /var/ossec/framework/python/bin/python3 /tmp/iplist-to-cdblist.py /var/ossec/etc/lists/alienvault_reputation.ipset /var/ossec/etc/lists/blacklist-alienvault`

```
jacob@jacob-virtual-machine:~$ sudo /var/ossec/framework/python/bin/python3 /tmp/iplist-to-cdblist.py /var/ossec/etc/lists/alienvault_reputation.ipset /var/ossec/etc/lists/blacklist-alienvault
jacob@jacob-virtual-machine:~$
```

I assigned the right permissions and ownership to the generated file:

- `sudo chown wazuh:wazuh /var/ossec/etc/lists/blacklist-alienvault`

Configured the Active Response module to block the malicious IP address and added a custom rule to trigger a Wazuh active response script in /var/ossec/etc/rules/local_rules.xml custom ruleset file:


```

GNU nano 6.2 /var/ossec/etc/rules/local.rules.xml
<!-- 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>
    <script>1.1.1.1</script>
    <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="attack">
  <rule id="100100" level="10">
    <if group=webattack|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>

```

Edited the Wazuh server `/var/ossec/etc/ossec.conf` configuration file and added the `etc/lists/blacklist-alienvault` list to the `<ruleset>` section:

```

<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>

```

Added the Active Response block to the Wazuh server `/var/ossec/etc/ossec.conf` file:

```

<ossec_config>
  <active-response>
    <command>firewall-drop</command>
    <location>local</location>
    <rules_id>100100</rules_id>
    <timeout>60</timeout>
  </active-response>
</ossec_config>

```

For the fedora endpoint: The `firewall-drop` command integrates with the fedora local iptables firewall and drops incoming network connection from the attacker endpoint for 60 seconds:

I restarted the Wazuh manager to apply the changes:

- `sudo systemctl restart wazuh-manager`

I accessed the web server from the RHEL endpoint using the IP address from the attacker endpoint:

- `curl http://192.168.47.139`

```

You should now put your content in a location of your choice and
edit the root configuration directive in the Caddy configuration
file <code>etc/caddy/Caddyfile</code>.

<div id="logo">
  <a href="https://getfedora.org/" id="fedora-poweredby"><a> Fedora -->
 <!-- subserver -->
</div>
</div>

<footer class="col-sm-12">
  <a href="https://apache.org">Apache</a> is a registered trademark o
f <a href="https://apache.org">the Apache Software Foundation</a> in the United
States and/or other countries. <br />
  <a href="https://nginx.org">NGINX</a> is a registered trademark of
<a href="https://75 Networks, Inc.</a>
  <a href="https://caddyserver.com">Caddy</a> is a registered tradema
rk of Stack Holdings GmbH.
</footer>

</body>
</html>
root00PMsense:" ■

```

Here, the attacker endpoint connects to the victim's web servers for the first time. After the first connection, the Wazuh Active Response module temporarily blocks any successive connection to the web servers for 60 seconds.

I visualized the alert data in the Wazuh dashboard

Mar 2, 2025 @ 00:06:36.378	fedora	IP address found in AlienVault reputation database.	10	100/100
Mar 2, 2025 @ 00:05:10.180	fedora	IP address found in AlienVault reputation database.	10	100/100
Mar 2, 2025 @ 23:57:59.622	fedora	IP address found in AlienVault reputation database.	10	100/100
Mar 2, 2025 @ 23:56:37.476	fedora	IP address found in AlienVault reputation database.	10	100/100
Mar 2, 2025 @ 23:54:55.383	fedora	IP address found in AlienVault reputation database.	10	100/100

Document Details [View surrounding documents](#) [View single document](#)

Table	JSON
..index	wazuh-alerts-4.x-2025.03.02
agent.id	002
agent.ip	192.168.47.142
agent.name	fedora
data.id	403
data.protocol	GET
data.srcip	192.168.47.139
data.url	/
decoder.name	web-accesslog
full_log	192.168.47.139 - - [03/Mar/2025:00:06:35 +0100] "GET / HTTP/1.1" 403 8474 "-" "curl/8.12.1"
id	1740956796.21599
input.type	log
location	/var/log/httpd/access_log
manager.name	jacob-virtual-machine
rule.description	IP address found in AlienVault reputation database.
rule.firedtimes	2
rule.groups	attack
rule.id	100100
rule.level	10

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.

I performed the following steps to configure the fedora endpoint. This allows performing authentication failure attempts on the monitored RHEL and Windows endpoints.

- On the attacker endpoint, I installed Hydra and used it to execute the brute-force attack:
 - sudo apt update
 - sudo apt install -y hydra

1. I created a text file with 10 random passwords

```

GNU nano 6.2 pass.text
123
456
789
123
456
789
123
456
789
123

```

2. I run Hydra from the attacker endpoint to execute brute-force attacks against the 3 endpoints separately. example:

- `sudo hydra -l badguy -P pass.text 192.168.47.139 ssh`

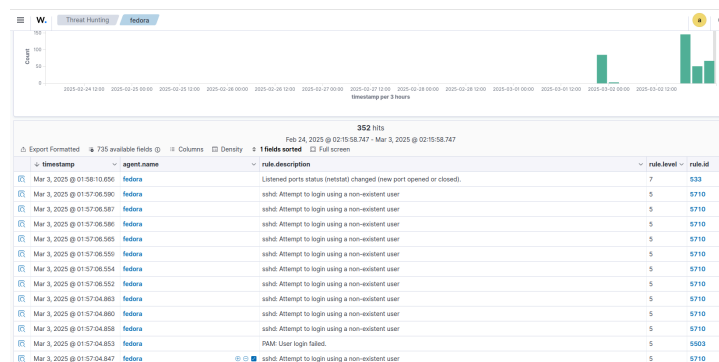
```

C:\Users\jacob>hydra -l badguy -P pass.text 192.168.47.142 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 2025-03-03 01:57:02
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 11 tasks per 1 server, overall 11 tasks, 11 login tries (1:3/pull), ~1 try per task
[DATA] attacking ssh://192.168.47.142:22/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-03-03 01:57:07

```

I visualized the alerts:



Task 5 - SOC integrations <select any 1>

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 created the Docker repository file and **updated the package lists, and verified that the repo exists:**

```

jacobakor@fedora:~$ sudo tee /etc/yum.repos.d/docker-ce.repo <<EOF
[docker-ce-stable]
name=Docker CE Stable - $basearch
baseurl=https://download.docker.com/linux/fedora/$releasever/$basearch/stable
enabled=1
gpgcheck=1
gpgkey=https://download.docker.com/linux/fedora/gpg
EOF
[docker-ce-stable]
name=Docker CE Stable - $basearch
baseurl=https://download.docker.com/linux/fedora/$releasever/$basearch/stable
enabled=1
gpgcheck=1
gpgkey=https://download.docker.com/linux/fedora/gpg
jacobakor@fedora:~$ sudo dnf repolist
repo id          repo name
docker-ce-stable Docker CE Stable - x86_64
fedora            Fedora 41 - x86_64
fedora-cisco-openh264 Fedora 41 openh264 (From Cisco) - x86_64
updates          Fedora 41 - x86_64 - Updates
wazuh             Wazuh repository

```

Then, I installed Docker

- `sudo dnf install -y docker-ce docker-ce-cli containerd.io docker-compose-plugin`

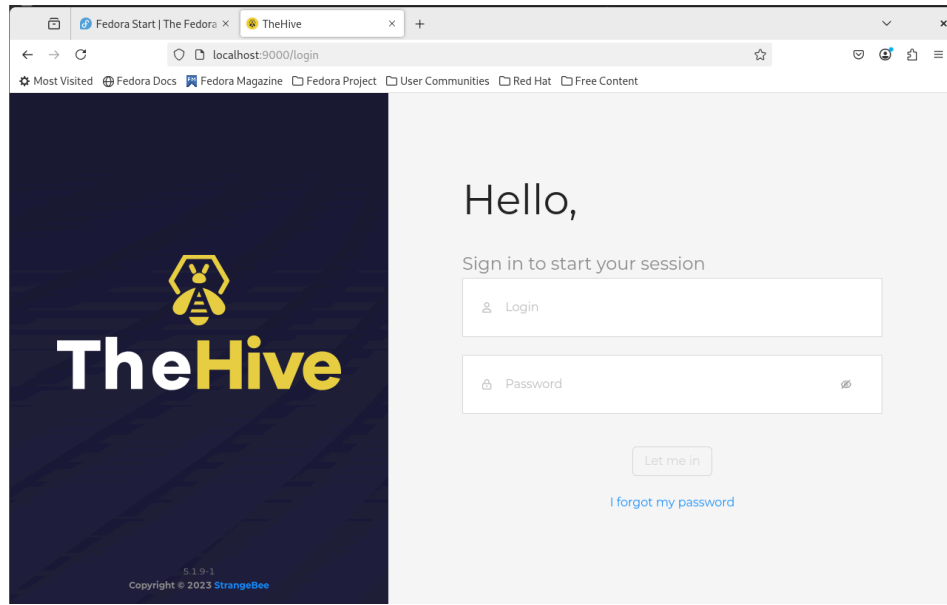
```
jacobakor@fedora:~$ sudo dnf install -y docker-ce docker-ce-cli containerd.io docker-compose-plugin
Updating and loading repositories:
Docker CE Stable - x86_64
Repositories loaded.
Package Arch Version Repository Size
Installing:
containerd.io x86_64 1.7.25-3.1.fc41 docker-ce-stable 150.5 MiB
docker-ce x86_64 3:28.0.1-1.fc41 docker-ce-stable 83.1 MiB
docker-ce-cli x86_64 3:28.0.1-1.fc41 docker-ce-stable 33.7 MiB
docker-compose-plugin x86_64 2.33.1-1.fc41 docker-ce-stable 70.9 MiB
Installing dependencies:
libgroup x86_64 3.0-6.fc41 fedora 157.8 KiB
slirpnetns x86_64 1.3.1-1.fc41 updates 93.4 KiB
Installing weak dependencies:
docker-buildx-plugin x86_64 0.21.1-1.fc41 docker-ce-stable 73.3 MiB
docker-ce-rootless-extras x86_64 28.0.1-1.fc41 docker-ce-stable 10.4 MiB
Transaction Summary:
Installing: 8 packages
Total size of inbound packages is 185 MiB. Need to download 185 MiB.
After this operation, 422 MiB extra will be used (install 422 MiB, remove 0 B).
(1/8) docker-ce-cli-3:28.0.1-1.fc41.x86_64 100% | 1.1 MiB/s | 8.2 MiB | 00m01s
(2/8) docker-ce-3:28.0.1-1.fc41.x86_64 100% | 2.3 MiB/s | 19.7 MiB | 00m08s
(3/8) libgroup-3.0-6.fc41.x86_64 100% | 15.3 KiB/s | 73.0 KiB | 00m05s
(4/8) docker-compose-plugin-2.33.1-1.fc41.x86_64 100% | 2.2 MiB/s | 34.6 MiB | 00m15s
(5/8) docker-ce-rootless-extras-28.0.1-1.fc41.x86_64 100% | 2.3 MiB/s | 3.2 MiB | 00m01s
(6/8) slirpnetns-1.3.1-1.fc41.x86_64 100% | 12.3 KiB/s | 47.1 KiB | 00m04s
(7/8) containerd.io-1.7.25-3.1.fc41.x86_64 100% | 2.2 MiB/s | 42.7 MiB | 00m19s
(8/8) docker-buildx-plugin-0.21.1-1.fc41.x86_64 100% | 3.2 MiB/s | 16.3 MiB | 00m05s
-----
(9/8) Total 100% | 5.2 MiB/s | 104.8 MiB | 00m25s
(1/9) https://download.docker.com/linux/fedora/gpg 100% | 3.3 KiB/s | 1.6 KiB | 00m00s
(2/9) Total 100% | 5.2 MiB/s | 104.8 MiB | 00m25s
Importing GPG key 0x621E9F35:
UserID : "Docker Release (CE rpm) <docker@docker.com>"
Fingerprint: 0000c11252a37f7d207f4c32f8b8621e9f35
From : https://download.docker.com/linux/fedora/gpg
The key was successfully imported.
(1/10) Verify package files 100% | 2.0 B/s | 8.0 B | 00m04s
(2/10) Prepare transaction 100% | 1.0 B/s | 8.0 B | 00m04s
(3/10) Installing slirpnetns-1.3.1-1.fc41.x86_64 100% | 551.6 KiB/s | 94.8 KiB | 00m00s

(4/10) Installing libgroup-3.0-6.fc41.x86_64 100% | 1.2 MiB/s | 150.2 KiB | 00m00s
(5/10) Installing containerd.io-1.7.25-3.1.fc41.x86_64 100% | 17.0 MiB/s | 150.9 MiB | 00m08s
(6/10) Installing docker-ce-cli-3:28.0.1-1.fc41.x86_64 100% | 12.9 MiB/s | 33.7 MiB | 00m03s
(7/10) Installing docker-ce-3:28.0.1-1.fc41.x86_64 100% | 17.1 MiB/s | 83.1 MiB | 00m05s
(8/10) Installing docker-ce-rootless-extras-28.0.1-1.fc41.x86_64 100% | 46.2 MiB/s | 10.4 MiB | 00m00s
(9/10) Installing docker-buildx-plugin-0.21.1-1.fc41.x86_64 100% | 69.1 MiB/s | 73.3 MiB | 00m01s
(10/10) Installing docker-compose-plugin-2.33.1-1.fc41.x86_64 100% [=====] | 13.7 MiB/s | 70.9 MiB | 00m00s
warning: posix.wait(): .fork(), .exec(), .wait() and .redirect2null() are deprecated, use rpm.spawn() or rpm.execute() instead
(10/10) Installing docker-compose-plugin-2.33.1-1.fc41.x86_64 100% | 7.9 MiB/s | 70.9 MiB | 00m00s
complete
```

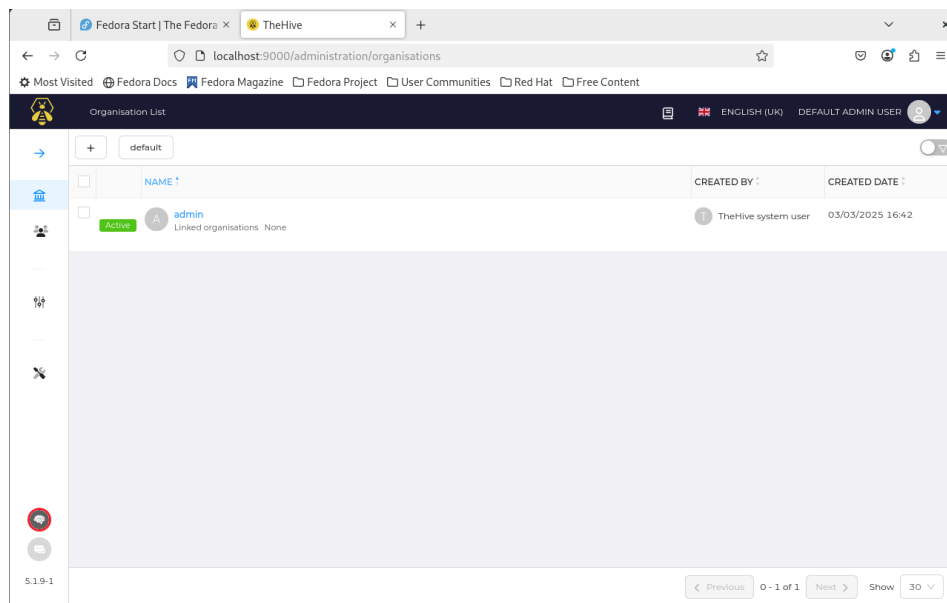
sudo docker compose up -d

```
jacobakor@fedora:~$ docker compose up -d
[00m00m00] /home/jacobakor/docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
✔ thehive Pulled 123.3s
✔ 759780526b78 Pull complete 19.0s
✔ 18c444e2ce Pull complete 19.0s
✔ 70b751aa3a Pull complete 14.1s
✔ f6aa1a0fed55 Pull complete 114.0s
✔ c3a767a7047 Pull complete 121.0s
✔ fc187ea32c Pull complete 121.0s
✔ 3318c1fa909 Pull complete 121.0s
✔ 1cf59ae37008 Pull complete 122.0s
✔ 59a42e2913 Pull complete 122.0s
✔ adbb8dc1c94 Pull complete 122.0s
✔ 1f90ff0ba85 Pull complete 122.0s
✔ 452baa89cc Pull complete 122.1s
✔ elasticsearch Pulled 194.0s
✔ 18859344005 Pull complete 191.0s
✔ 40294841a76 Pull complete 111.0s
✔ 7137590c5e4f Pull complete 111.0s
✔ e40f96c550d5 Pull complete 189.0s
✔ a27a129535bd Pull complete 189.0s
✔ 89507bf76e2 Pull complete 194.0s
✔ 9e4c9b8013f Pull complete 194.0s
✔ 484a9e31a41 Pull complete 194.0s
✔ b596c848dae4 Pull complete 194.0s
✔ Network jacobakor_default Created 0.7s
✔ Volume "jacobakor_esdata" Created 0.2s
✔ Volume "jacobakor_thehive_data" Created 0.2s
✔ Container elasticsearch Started 1.3s
✔ Container thehive Started 1.0s
jacobakor@fedora:~$
```

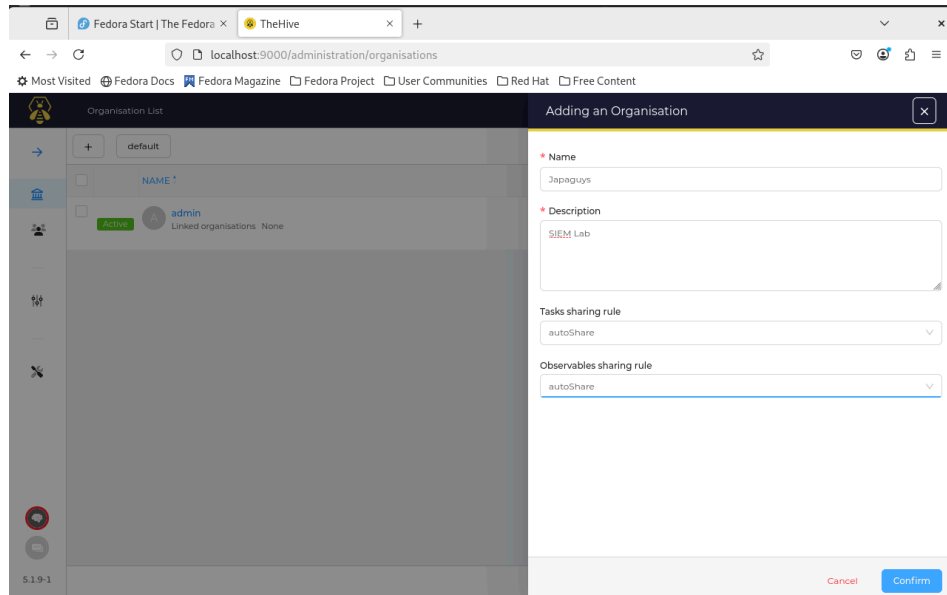
It was possible for me to access TheHive website



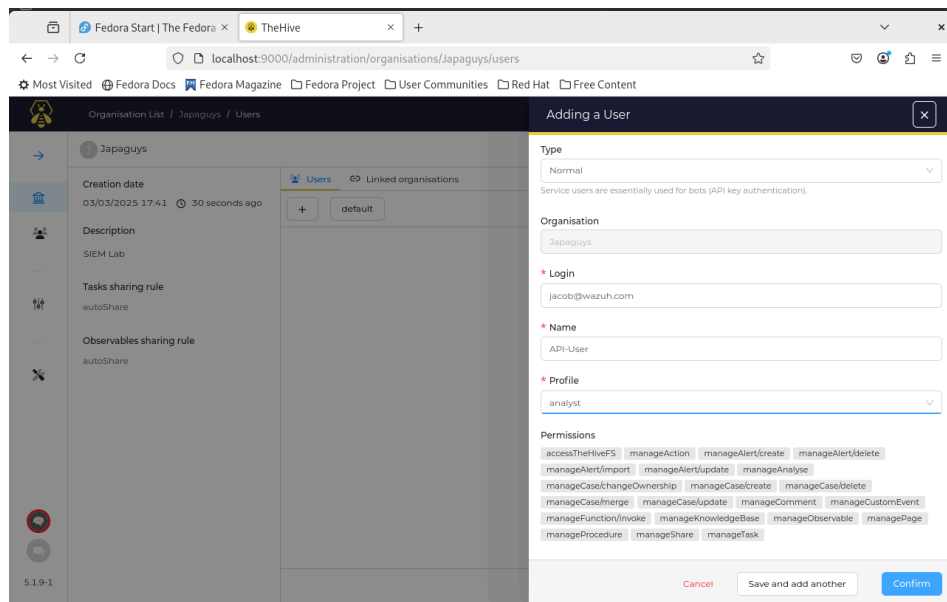
Then, I logged into the TheHive website the default credentials



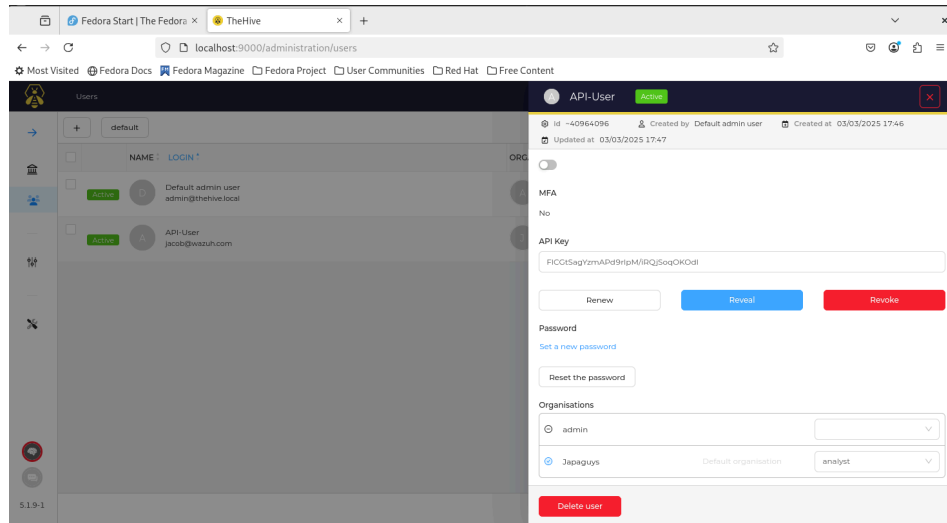
I proceeded by adding an organization:



I added a user:

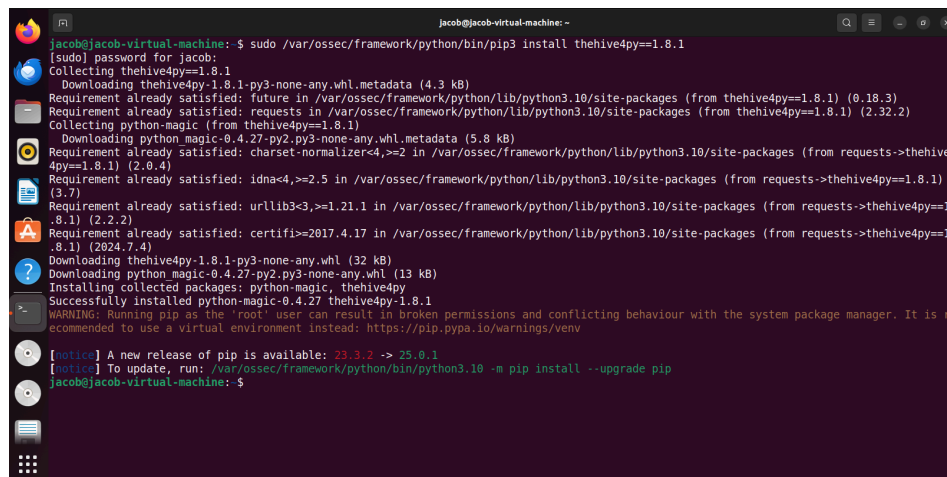


Created the API Key



I installed first, TheHive Python Module using the command:

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



Created the custom integration script by pasting the following python code in `/var/ossec/integrations/custom-w2hive.py`.

The `lvl_threshold` variable in the script indicates the minimum alert level that will be forwarded to TheHive. The variable can be customized so that only relevant alerts are forwarded to TheHive:

```
#!/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
```

```

#start user config

# Global vars

#threshold for wazuh rules level
lvl_threshold=0
#threshold for suricata rules level
suricata_lvl_threshold=3

debug_enabled = False
#info about created alert
info_enabled = True

#end user config

# Set paths
pwd = os.path.dirname(os.path.dirname(os.path.realpath(__file__)))
log_file = '{0}/logs/integrations.log'.format(pwd)
logger = logging.getLogger(__name__)
#set logging level
logger.setLevel(logging.WARNING)
if info_enabled:
    logger.setLevel(logging.INFO)
if debug_enabled:
    logger.setLevel(logging.DEBUG)
# create the logging file handler
fh = logging.FileHandler(log_file)
formatter = logging.Formatter('%(asctime)s - %(name)s - %(levelname)s - %(message)s')
fh.setFormatter(formatter)
logger.addHandler(fh)

def main(args):
    logger.debug('#start main')
    logger.debug('#get alert file location')
    alert_file_location = args[1]
    logger.debug('#get TheHive url')
    thive = args[3]
    logger.debug('#get TheHive api key')
    thive_api_key = args[2]
    thive_api = TheHiveApi(thive, thive_api_key )
    logger.debug('#open alert file')
    w_alert = json.load(open(alert_file_location))
    logger.debug('#alert data')
    logger.debug(str(w_alert))
    logger.debug('#gen json to dot-key-text')
    alt = pr(w_alert, ',', [])
    logger.debug('#formatting description')
    format_alt = md_format(alt)
    logger.debug('#search artifacts')
    artifacts_dict = artifact_detect(format_alt)
    alert = generate_alert(format_alt, artifacts_dict, w_alert)
    logger.debug('#threshold filtering')

```



```

if w_alert['rule']['groups']==['ids','suricata']:
    #checking the existence of the data.alert.severity field
    if 'data' in w_alert.keys():
        if 'alert' in w_alert['data']:
            #checking the level of the source event
            if int(w_alert['data']['alert']['severity'])<=suricata_lvl_threshold:
                send_alert(alert, thive_api)
        elif int(w_alert['rule']['level'])>=lvl_threshold:
            #if the event is different from suricata AND suricata-event-type: alert check lvl_threshold
            send_alert(alert, thive_api)

def pr(data,prefix, alt):
    for key,value in data.items():
        if hasattr(value,'keys'):
            pr(value,prefix+'.'+str(key),alt=alt)
        else:
            alt.append((prefix+'.'+str(key)+'|||'+str(value)))
    return alt

def md_format(alt,format_alt=''):
    md_title_dict = {}
    #sorted with first key
    for now in alt:
        now = now[1:]
        #fix first key last symbol
        dot = now.split('|||')[0].find('.')
        if dot==-1:
            md_title_dict[now.split('|||')[0]] =[now]
        else:
            if now[0:dot] in md_title_dict.keys():
                (md_title_dict[now[0:dot]]).append(now)
            else:
                md_title_dict[now[0:dot]]= [now]
    for now in md_title_dict.keys():
        format_alt+= '### '+now.capitalize()+'\n'+ ' | key | val |\n| ----- | ----- |\n'
        for let in md_title_dict[now]:
            key,val = let.split('|||')[0],let.split('|||')[1]
            format_alt+= ' | **' + key + '** | ' + val + ' |\n'
    return format_alt

def artifact_detect(format_alt):
    artifacts_dict = {}
    artifacts_dict['ip'] = re.findall(r'\d+\.\d+\.\d+\.\d+',format_alt)
    artifacts_dict['url'] = re.findall(r'http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|[*\(\)\,])|(?:%[0-9a-fA-F][0-9a-fA-F]))+',format
_alt)
    artifacts_dict['domain'] = []
    for now in artifacts_dict['url']: artifacts_dict['domain'].append(now.split('/')[1].split('/')[0])
    return artifacts_dict

```

```

def generate_alert(format_alt, artifacts_dict,w_alert):
    #generate alert sourceRef
    sourceRef = str(uuid.uuid4())[0:6]
    artifacts = []
    if 'agent' in w_alert.keys():
        if 'ip' not in w_alert['agent'].keys():
            w_alert['agent']['ip']='no agent ip'
        else:
            w_alert['agent'] = {'id':'no agent id', 'name':'no agent name'}

    for key,value in artifacts_dict.items():
        for val in value:
            artifacts.append(AlertArtifact(dataType=key, data=val))
    alert = Alert(title=w_alert['rule']['description'],
        tlp=2,
        tags=['wazuh',
            'rule='+w_alert['rule']['id'],
            'agent_name='+w_alert['agent']['name'],
            'agent_id='+w_alert['agent']['id'],
            'agent_ip='+w_alert['agent']['ip'],],
        description=format_alt ,
        type='wazuh_alert',
        source='wazuh',
        sourceRef=sourceRef,
        artifacts=artifacts,)
    return alert

def send_alert(alert, thrive_api):
    response = thrive_api.create_alert(alert)
    if response.status_code == 201:
        logger.info('Create TheHive alert: ' + str(response.json()['id']))
    else:
        logger.error('Error create TheHive alert: {}/{}'.format(response.status_code, response.text))

if __name__ == "__main__":
    try:
        logger.debug('debug mode') # if debug enabled
        # Main function
        main(sys.argv)
    except Exception:
        logger.exception('EGOR')

```

Created a bash script as `/var/ossec/integrations/custom-w2thive` . This will properly execute the .py script created in the previous step:

```

#!/bin/sh
# 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="$(cd $(dirname ${SCRIPT_PATH_NAME}); pwd -P)"
SCRIPT_NAME="$(basename ${SCRIPT_PATH_NAME})"

case ${DIR_NAME} in
    */active-response/bin | */wodles*)
        if [ -z "${WAZUH_PATH}" ]; then
            WAZUH_PATH="$(cd ${DIR_NAME}/../.; pwd)"
        fi

        PYTHON_SCRIPT="${DIR_NAME}/${SCRIPT_NAME}.py"
        ;;
    */bin)
        if [ -z "${WAZUH_PATH}" ]; then
            WAZUH_PATH="$(cd ${DIR_NAME}/../; pwd)"
        fi

        PYTHON_SCRIPT="${WAZUH_PATH}/framework/scripts/${SCRIPT_NAME}.py"
        ;;
    */integrations)
        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} $@

```

To allow Wazuh to run the integration script, I added the following lines to the manager configuration file at `/var/ossec/etc/ossec.conf`. Inserted the IP address for the TheHive server along with the API key that was generated earlier:

```

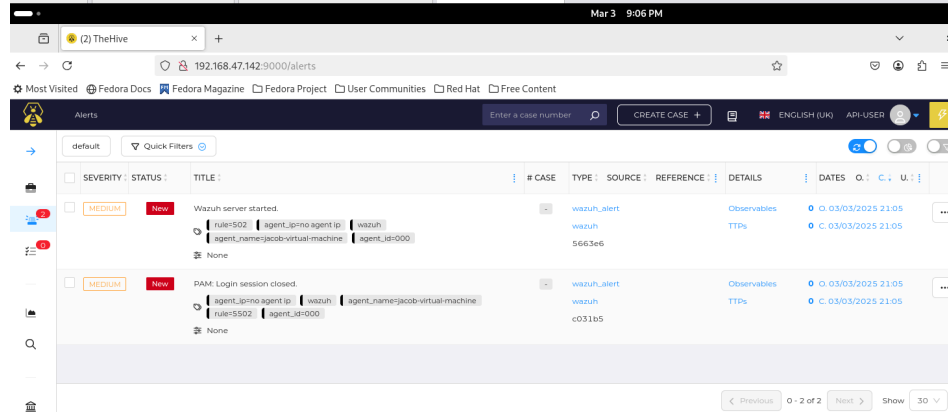
<ossec_config>
  <integration>
    <name>custom-w2thive</name>
    <hook_url>http://192.168.47.142:9000</hook_url>
    <api_key>FLCGtSagYzmAPd9rLpM/iRQjSoq0K0dI</api_key>
    <alert_format>json</alert_format>
  </integration>
</ossec_config>

```

Then, restart the manager to apply the changes:

- `sudo systemctl restart wazuh-manager`

I logged into TheHive with my test user account, and i can see Wazuh generated alerts under the "Alerts" tab



References:

<https://wazuh.com/blog/using-wazuh-and-thehive-for-threat-protection-and-incident-response/>

<https://documentation.wazuh.com/current/index.html>

<https://documentation.wazuh.com/current/proof-of-concept-guide/detect-brute-force-attack.html>