

SOC Automation Lab: Integrating Wazuh, TheHive, Shuffle, and VirusTotal



WAZUH



THEHIVE



SHUFFLE



VIRUSTOTAL

SOC Automation Lab: Integrating Wazuh, TheHive, Shuffle, and VirusTotal

Lab Overview

This lab demonstrates how Security Operations Center (SOC) tasks can be automated by integrating multiple open-source tools. The goal is to reduce manual workload, improve detection and response speed, and provide security analysts with streamlined alerts.

Objectives

By the end of this lab, a user will be able to:

- Monitor endpoints and servers using **Wazuh SIEM**
- Create and customize **rules and indexes** in Wazuh
- Add and manage **Wazuh agents** for log collection
- Detect suspicious activities (e.g., **Mimikatz usage**)

- Extract and analyze **file hashes (SHA-256)** from Wazuh logs
- Automatically submit hashes to **VirusTotal** for reputation checks
- Trigger automated workflows in **Shuffle** for alert enrichment
- Forward critical alerts to **TheHive** for case management
- Send **email notifications** to administrators when incidents are detected

Prerequisites

Before you begin, ensure you have the following environment and tools ready:

Local Environment

- **Host OS:** Windows 10/11
- **Virtualization:** VMware Workstation or VMware Player
- **Windows 11 VM:** Used as the endpoint machine (Sysmon + Wazuh agent installed)

Cloud Environment (DigitalOcean)

- **Ubuntu VM #1:** Wazuh SIEM Server
- **Ubuntu VM #2:** TheHive (Incident Response Platform)

Tools & Services

- **Sysmon** (installed on Windows 11 VM for detailed event logging)
- **Wazuh Agent** (installed on Windows 11 VM for log forwarding)
- **Shuffle** (cloud-hosted, used for automation workflows)
- **VirusTotal** (Free API key for hash reputation checks)
- **Strangebee License** (Free Trial)
 - Registration on StrangeBee requires a valid **work** or **student** email address.

Learning Outcomes

During the development of this lab, I gained hands-on experience with:

- Deploying and configuring **Wazuh** for log analysis
- Creating custom **detection rules and indexes**
- Building **automation workflows** in Shuffle
- Ingesting alerts into **TheHive** for structured incident response
- Integrating third-party threat intelligence (**VirusTotal**) into the SOC pipeline

Diagram

Setting Up Windows 11 Virtual Machine

In this section, you'll set up a Windows 11 virtual machine, which will act as your monitored endpoint in the SOC Automation Lab.

1. Download Windows 11 ISO

1. Go to the official Microsoft download page:  [Download Windows 11 Disk Image \(ISO\)](#)
2. Under “**Download Windows 11 Disk Image (ISO) for x64 devices**”, select:
3. **Windows 11 (multi-edition ISO for x64 devices)** → click **Confirm**
4. Choose **Language** → *English (United States)* → click **Confirm**
5. Select **64-bit Download** → the ISO file download will begin.

2. Create Windows 11 VM in VMware

1. Open **VMware Workstation / Player**.
2. Click **Create a New Virtual Machine**.
3. Select **Typical (recommended)** → click **Next**.
4. Choose **Installer disc image file (ISO)** and browse to the Windows 11 ISO you downloaded → click **Next**.

5. Enter a username and password (make sure to remember it, you'll need it later). → click **Next**.
6. Specify the **disk size** (use the recommended value) → click **Next**.
7. (Optional) Customize hardware settings (e.g., RAM, CPU, or network adapter). If unsure, the default configuration works fine.
8. Check **Power on this virtual machine after creation**.
9. Click **Finish**.

3. Complete Windows 11 Installation

1. The VM will start and boot from the ISO.
2. Follow the on-screen installation steps (language, edition, setup options).
3. Once installation is complete, log in with the password you created earlier.

Now you have a working **Windows 11 VM** inside VMware. This will be used later for deploying and testing SOC automation tools (Wazuh, TheHive, Shuffle).

Installing Sysmon on Windows 11 VM

In this step, you'll install Sysmon on your Windows 11 VM to generate rich security logs that Wazuh will later monitor and analyze.

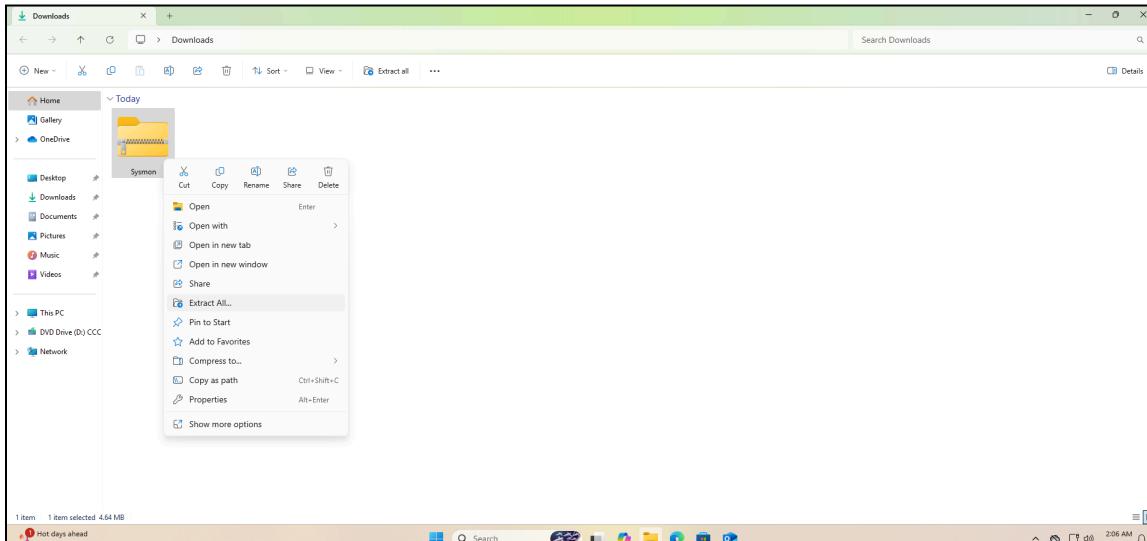
Why Sysmon?

Sysmon (System Monitor) is a Windows system service and device driver from the Sysinternals suite. It monitors and logs system activity, such as process creation, network connections, and file changes. In this project, you will use Sysmon to generate detailed security logs that can later be ingested by **Wazuh** for detection and automated response.

1. Download Sysmon

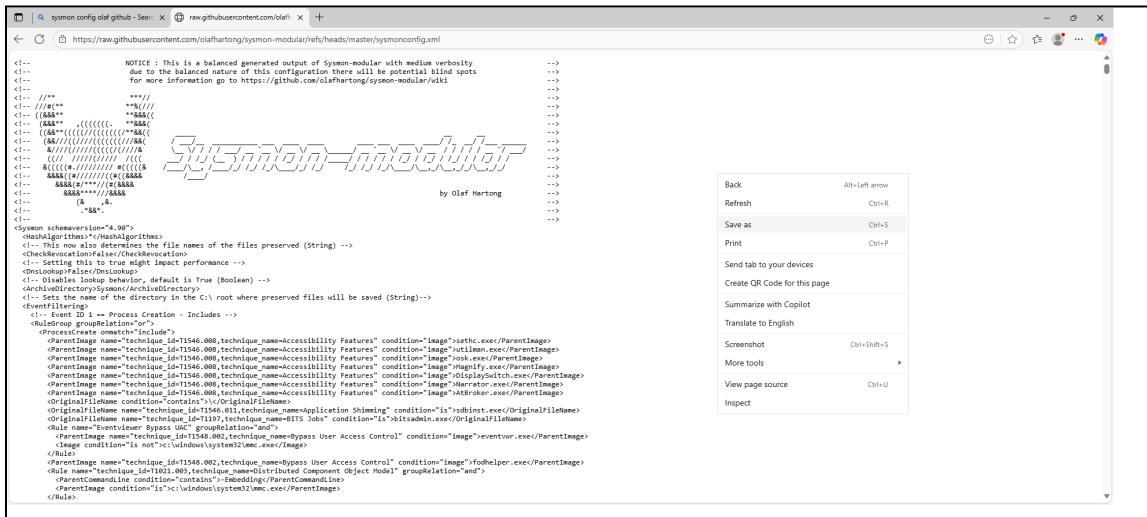
1. Inside your Windows 11 VM, open **Microsoft Edge**.
2. Go to the official Sysmon download page:  [Sysmon – Windows Sysinternals](#)
3. Click **Download Sysmon**.
4. Once the download is complete, go to the download location.

- Right-click the Sysmon archive and select **Extract All** to unzip it.



2. Download Sysmon Configuration

- Go to the olafhartong Sysmon Modular configuration on GitHub: [Sysmon Modular Configuration \(olafhartong\)](https://github.com/olafhartong/sysmon-modular/blob/master/sysmonconfig.xml)
- Right-click on the page → **Save As...**

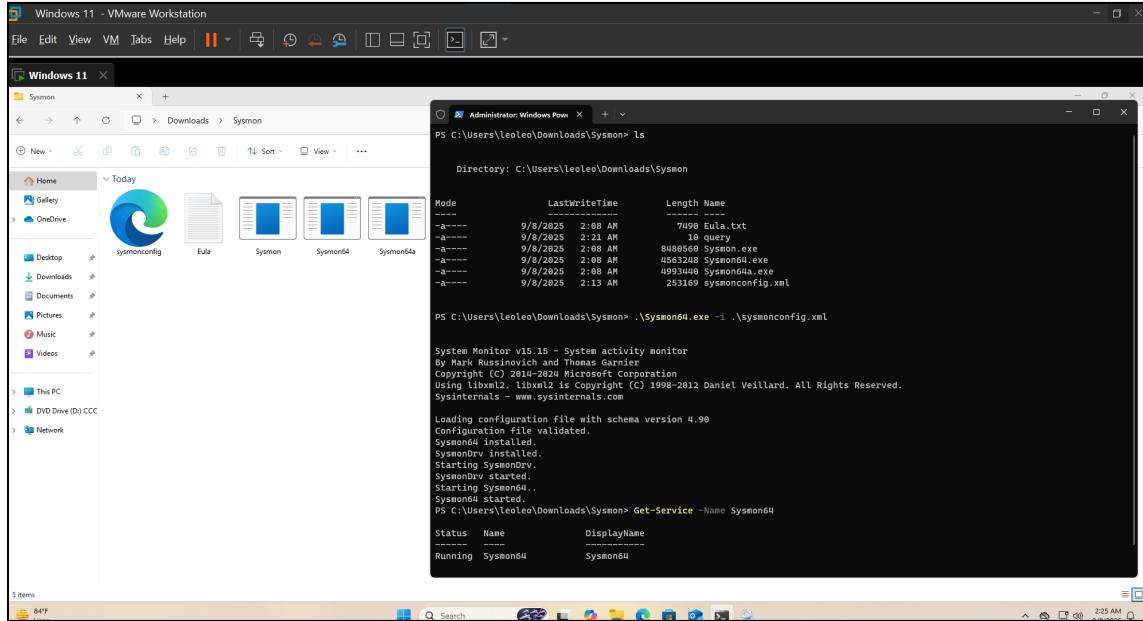


- Name the file **sysmonconfig.xml**
- Save it in the same Sysmon folder you extracted earlier.

3. Install Sysmon with Configuration

- Open **PowerShell as Administrator**.
- Navigate to the Sysmon directory: **cd "C:\path\to\Sysmon"**

- Run the installation command: `.\Sysmon64.exe -i sysmonconfig.xml`

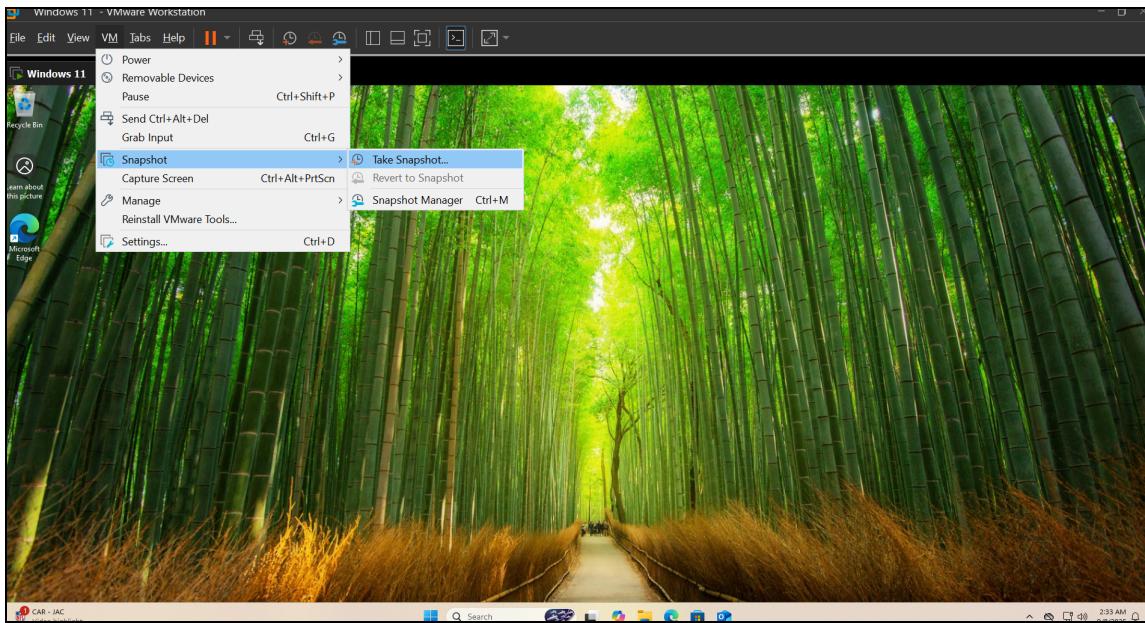


- Accept the license agreement.
- After installation, you should see the message: **Sysmon64 started.**
- Run the following command to confirm Sysmon is running: `Get-Service -Name sysmon64`
- If the **Status** shows **Running**, Sysmon has been successfully installed.
- To verify logs are being generated, open Event Viewer → Applications and Services Logs → Microsoft → Windows → Sysmon → Operational.

4. Create a VMware Snapshot

It's a good practice to save your VM state after installing Sysmon.

1. In VMware, go to the top menu → **VM** → **Snapshot** → **Take Snapshot**.



2. Give your snapshot a descriptive name (e.g., *After Sysmon Installation*).
3. Click **OK**.

✓ You now have **Sysmon installed and running** with a modular configuration, and a **VM snapshot** saved for rollback if needed.

Setting Up Wazuh and TheHive on DigitalOcean

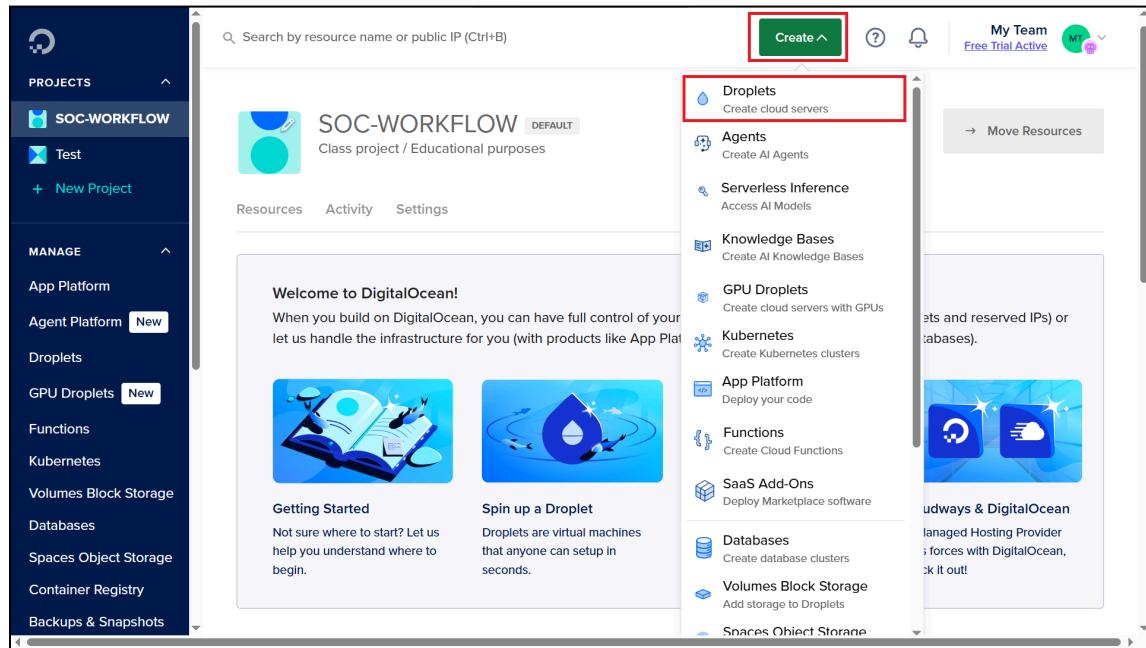
In this step, you'll deploy cloud-based Ubuntu servers for Wazuh and TheHive using DigitalOcean. These will act as your SOC's backend systems.

1. Sign Up for DigitalOcean

1. You can sign up here on DigitalOcean: <https://m.do.co/c/28868f21cdd3>
2. You can also use Vultr as an alternative: <https://www.vultr.com/register/>
3. Create an account using a **valid credit/debit card**.
 - DigitalOcean will temporarily charge **\$1 for verification**, which is refunded after setup.
4. After logging in, go to **Projects** → click **New Project**.
5. Give your project a **name** (e.g., SOC-Automation) and select its **purpose**.
6. Click **Create Project**.

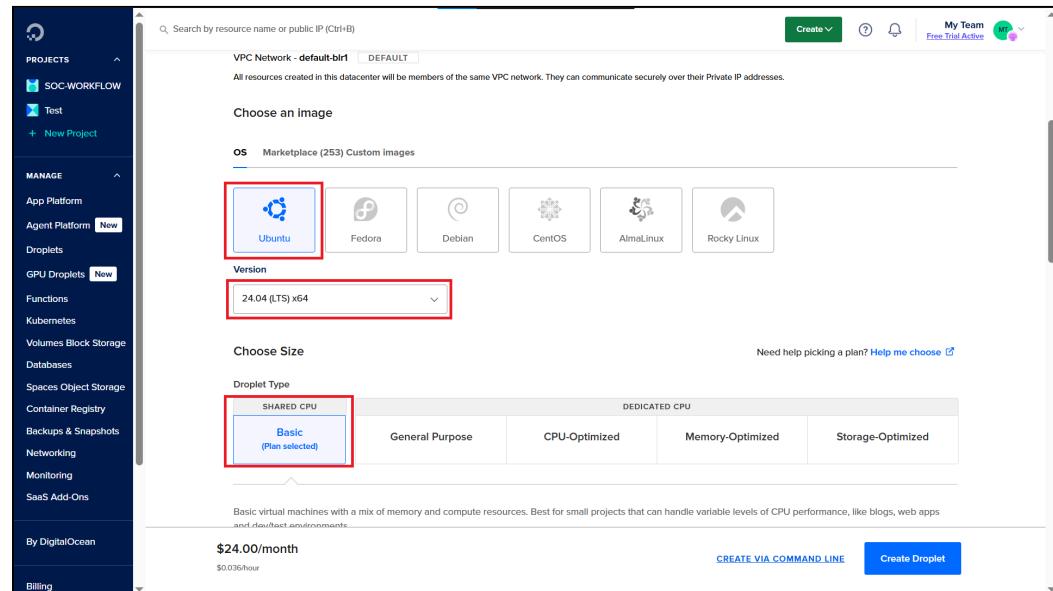
2. Create Wazuh Droplet

1. Inside your project, click **Create** → select **Droplets**.



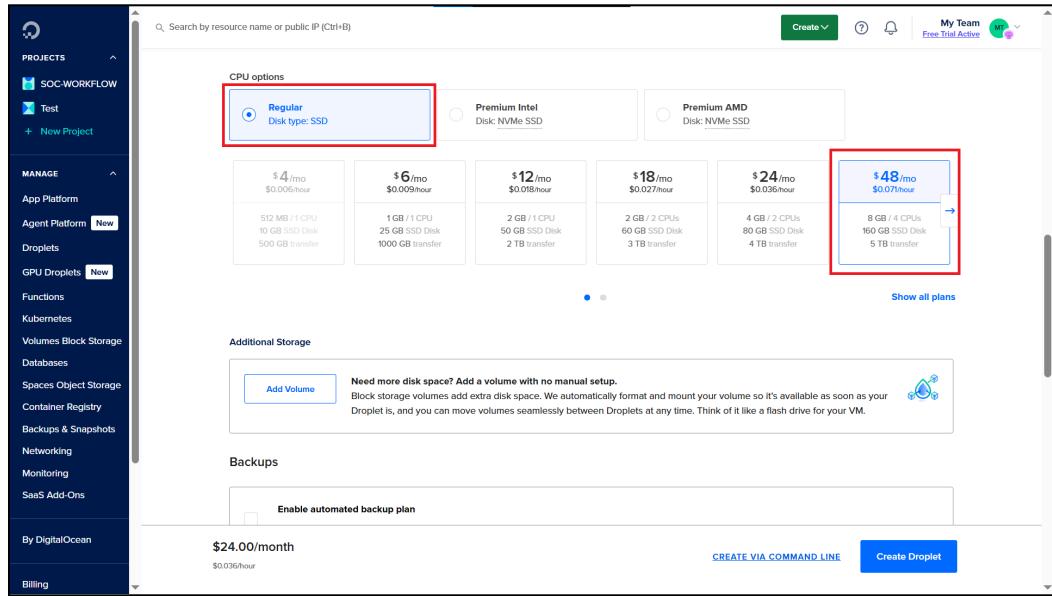
2. Configure your Droplet as follows:

a. **Image:** Ubuntu 24.04 (LTS) x64

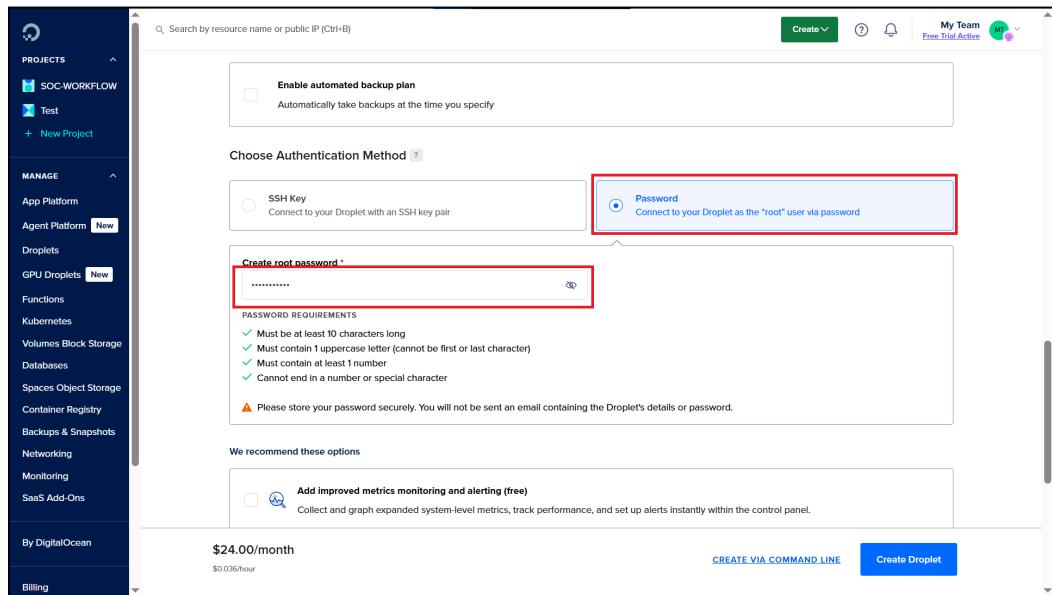


b. **CPU Options:** Regular

c. **Size:** 8 GB RAM / 4 vCPUs / 160 GB SSD Disk / 5 TB Transfer

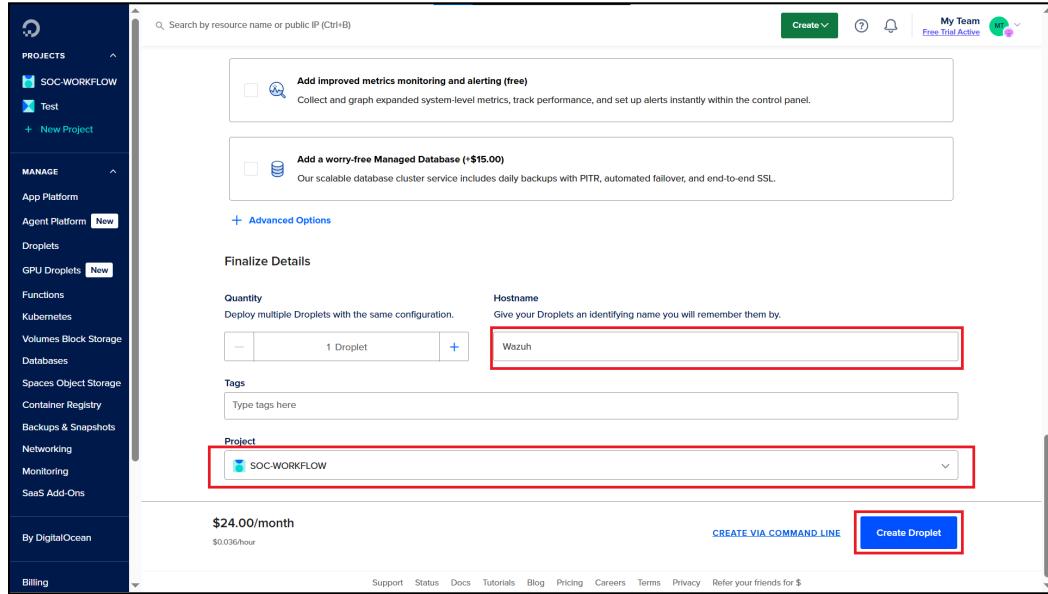


d. **Authentication:** Password (create a strong password and note it down)



- e. **Hostname:** Wazuh (you can choose any name, but functional names keep things organized)
- f. Select the project you created earlier.

g. Click **Create Droplet**.



h. To connect to your Droplet later, use: `ssh root@<droplet-ip>`

3. Create TheHive Droplet

Repeat the same steps as above, but this time:

- **Hostname: TheHive**
- **Recommended Size:** Ideally 16 GB RAM (TheHive is more resource-intensive than Wazuh).
 - If 16 GB isn't available in your plan, use the same specs as the Wazuh Droplet (8 GB RAM / 4 vCPUs).

4. Verify Droplets

- After creation, you will see both VMs listed under your project.

The screenshot shows the DigitalOcean control panel. On the left, a sidebar titled 'PROJECTS' contains a single item: 'SOC-WORKFLOW'. Under 'MANAGE', several options are listed: App Platform, Agent Platform (with a 'New' button), Droplets, GPU Droplets (with a 'New' button), Functions, Kubernetes, Volumes Block Storage, Databases, Spaces Object Storage, Container Registry, and Backups & Snapshots. The main content area is titled 'SOC-WORKFLOW' and describes it as a 'Class project / Educational purposes'. It shows 'Resources', 'Activity', and 'Settings' tabs. Under 'DROPLETS (2)', there are two entries: 'The-Hive' (IP: 68.183.84.53) and 'Wazuh' (IP: 139.59.25.255). Each entry has a 'Move Resources' button, an 'Upsize' button, and a '...' button. Below the droplet list, there's a section for creating new resources like AI inference endpoints, managed databases, GPU droplets, and spaces, along with links to product documentation, tutorials, and API/CLI docs.

- Note down the **public IP addresses** for **Wazuh** and **TheHive**.
- You will use these IPs later to connect via SSH and install the respective tools.

At this point, you have two Ubuntu VMs running on DigitalOcean:

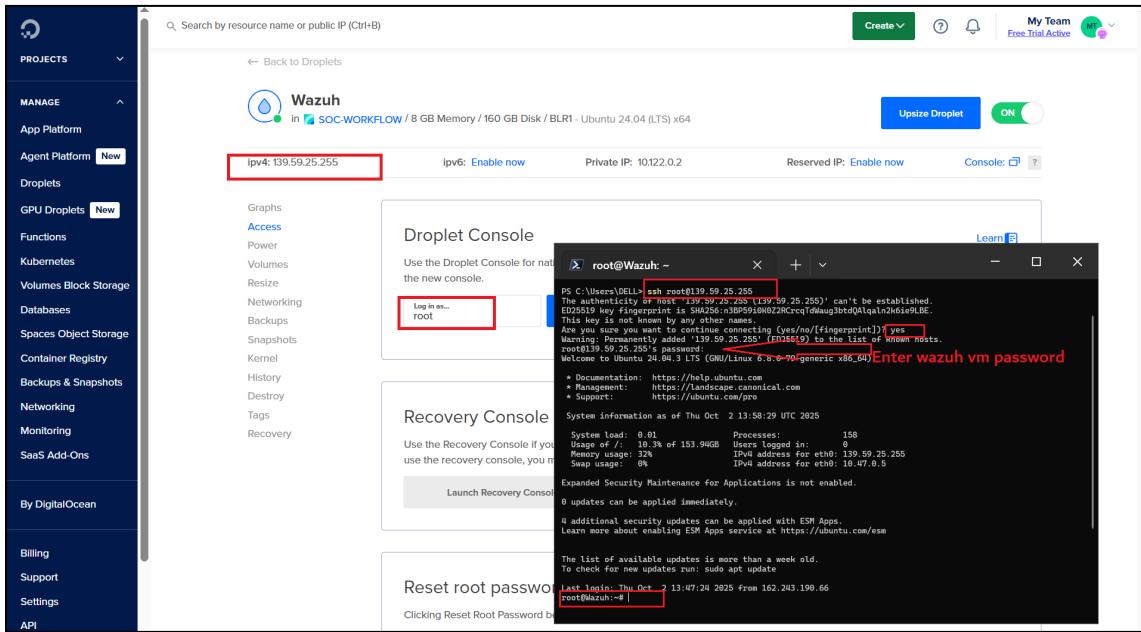
- Wazuh** → for SIEM setup
- TheHive** → for incident response and case management

Droplet	Purpose	Recommended Specs	Hostname
Wazuh	SIEM & log analysis	8 GB RAM / 4 vCPUs	Wazuh
The-Hive	Incident response	16 GB RAM / 4 vCPUs	The-Hive

Connect to the Wazuh VM

- Open **PowerShell** on your local machine.
- Run the following command (replace with your Wazuh VM IP): `ssh root@{your-wazuh-vm-ip}`
- Type **yes** and press **Enter**.

4. Enter the VM password you created during set



You are now inside your Wazuh VM.

Run the following to update your packages:

```
apt-get update && apt-get upgrade -y
```

Install Wazuh

1. Go to the official Wazuh installation guide: [Wazuh Quickstart](#)
2. Copy the **latest installation command** shown there.
3. Paste it into your Wazuh VM and press **Enter**.

```

root@Wazuh:~#
root@Wazuh:~# curl -s0 https://packages.wazuh.com/4.12/wazuh-install.sh && sudo bash ./wazuh-install.sh -a
08/09/2025 09:55:31 INFO: Starting Wazuh installation assistant. Wazuh version: 4.12.0
08/09/2025 09:55:31 INFO: Verbose logging redirected to /var/log/wazuh-install.log
08/09/2025 09:55:39 INFO: Verifying that your system meets the recommended minimum hardware requirements.
08/09/2025 09:55:39 INFO: Wazuh manager service port will be 443.
08/09/2025 09:55:39 INFO: --- Dependencies
08/09/2025 09:55:39 INFO: Installing debhelper.
08/09/2025 09:55:35 INFO: Wazuh repository added.
08/09/2025 09:55:35 INFO: --- Configuration files ---
08/09/2025 09:55:35 INFO: Generating configuration files.
08/09/2025 09:56:36 INFO: Generating the root certificate.
08/09/2025 09:56:36 INFO: Generating Admin certificates.
08/09/2025 09:56:37 INFO: Generating Wazuh indexer certificates.
08/09/2025 09:56:37 INFO: Generating Filebeat certificates.
08/09/2025 09:56:38 INFO: Generating Wazuh dashboard certificates.
08/09/2025 09:56:39 INFO: Created wazuh-install-files.tar. It contains the Wazuh cluster key, certificates, and passwords necessary for installation.
08/09/2025 09:56:39 INFO: --- Wazuh indexer ---
08/09/2025 09:56:39 INFO: Starting Wazuh indexer installation.
08/09/2025 09:57:06 INFO: Wazuh indexer installation finished.
08/09/2025 09:57:06 INFO: Wazuh indexer post-install configuration finished.
08/09/2025 09:57:06 INFO: Starting service wazuh-indexer.
08/09/2025 09:57:25 INFO: wazuh-indexer service started.
08/09/2025 09:57:25 INFO: Initializing Wazuh indexer cluster security settings.
08/09/2025 09:57:31 INFO: Wazuh indexer cluster security configuration initialized.
08/09/2025 09:57:31 INFO: Wazuh indexer cluster initialized.
08/09/2025 09:57:31 INFO: --- Wazuh server ---
08/09/2025 09:57:31 INFO: Starting the Wazuh manager installation.
08/09/2025 09:59:19 INFO: Wazuh manager installation finished.
08/09/2025 09:59:19 INFO: Wazuh manager vulnerability detection configuration finished.
08/09/2025 09:59:19 INFO: Starting service wazuh-manager.
08/09/2025 09:59:37 INFO: wazuh-manager service started.
08/09/2025 09:59:37 INFO: Starting Filebeat installation.
08/09/2025 09:59:45 INFO: Filebeat installation finished.
08/09/2025 09:59:46 INFO: Filebeat post-install configuration finished.
08/09/2025 09:59:46 INFO: Starting service filebeat.
08/09/2025 09:59:47 INFO: filebeat service started.
08/09/2025 09:59:47 INFO: --- Wazuh dashboard ---
08/09/2025 09:59:47 INFO: Starting Wazuh dashboard installation.
08/09/2025 10:00:54 INFO: Wazuh dashboard installation finished.
08/09/2025 10:00:54 INFO: Wazuh dashboard post-install configuration finished.
08/09/2025 10:00:54 INFO: Starting service wazuh-dashboard.
08/09/2025 10:00:55 INFO: wazuh-dashboard service started.
08/09/2025 10:00:57 INFO: Updating the internal users.
08/09/2025 10:01:02 INFO: A backup of the internal users has been saved in the /etc/wazuh-indexer/internalusers-backup folder.
08/09/2025 10:01:18 INFO: The filebeat.yml file has been updated to use the Filebeat Keystore username and password.
08/09/2025 10:01:52 INFO: Initializing Wazuh dashboard web application.
08/09/2025 10:01:52 INFO: Wazuh dashboard web application not yet initialized. Waiting...
08/09/2025 10:02:07 INFO: Wazuh dashboard web application not yet initialized. Waiting...
08/09/2025 10:02:22 INFO: Wazuh dashboard web application initialized.
08/09/2025 10:02:22 INFO: --- Summary ---
08/09/2025 10:02:22 INFO: You can access the web interface https://<wazuh-dashboard-ip>:443
User: admin
Password: qnQuixE=EU2XurDe.Uu.d.4PRS=GZ9Hr
08/09/2025 10:02:22 INFO: Installation finished.
root@Wazuh:~#

```

If everything is successful, you will see:

- A generated **user and password** → Write these down! You will need them for the Wazuh web dashboard.
- **Installation finished message**

Verify Wazuh Service

Check the status of the Wazuh manager service: `systemctl status wazuh-manager.service`

- If it's **running** → you're good to go.
- If not, restart the service and check again:

```

systemctl restart wazuh-manager.service
systemctl status wazuh-manager.service

```

```

root@Wazuh:~# systemctl status wazuh-manager.service
● wazuh-manager.service - Wazuh manager
   Loaded: loaded (/usr/lib/systemd/system/wazuh-manager.service; enabled; preset: enabled)
     Active: active (running) since Fri 2025-10-03 02:50:33 UTC; 1h 33min ago
       Process: 832 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (code=exited, status=0/SUCCESS)
      Tasks: 176 (limit: 9486)
     Memory: 2.5G (peak: 2.7G)
        CPU: 1min 58.231s
       CGroup: /system.slice/wazuh-manager.service
               ├─1235 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh_apid.py
               ├─1236 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh_apid.py
               ├─1237 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh_apid.py
               ├─1240 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh_apid.py
               ├─1243 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh_apid.py
               ├─1266 /var/ossec/bin/wazuh-integratord
               ├─1287 /var/ossec/bin/wazuh-authd
               ├─1300 /var/ossec/bin/wazuh-db
               ├─1326 /var/ossec/bin/wazuh-execd
               ├─1341 /var/ossec/bin/wazuh-analysisd
               ├─1353 /var/ossec/bin/wazuh-syscheckd
               ├─1367 /var/ossec/bin/wazuh-remoted
               ├─1449 /var/ossec/bin/wazuh-logcollector
               ├─1466 /var/ossec/bin/wazuh-monitord
               └─1479 /var/ossec/bin/wazuh-modulesd

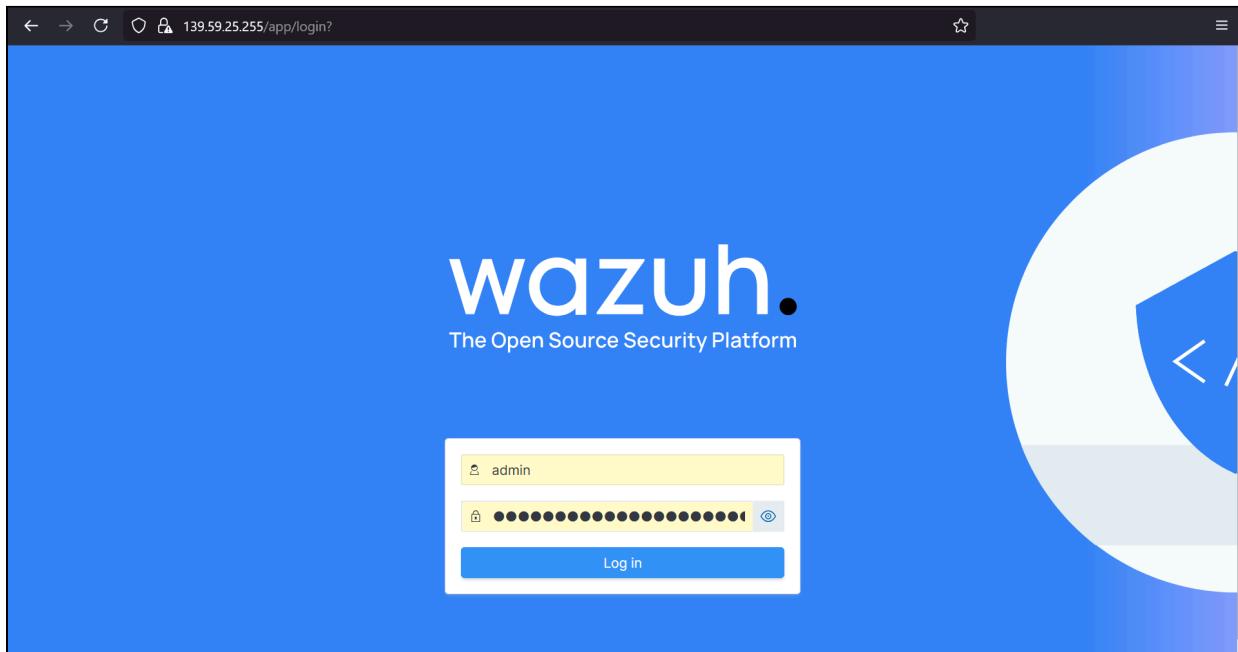
Oct 03 02:50:28 Wazuh env[832]: Started wazuh-analysisd...
Oct 03 02:50:28 Wazuh env[832]: Started wazuh-syscheckd...
Oct 03 02:50:29 Wazuh env[832]: Started wazuh-remoted...
Oct 03 02:50:29 Wazuh env[832]: Started wazuh-logcollector...
Oct 03 02:50:30 Wazuh env[832]: Started wazuh-monitord...
Oct 03 02:50:30 Wazuh env[1475]: 2025/10/03 02:50:30 wazuh-modulesd:router: INFO: Loaded router module.
Oct 03 02:50:30 Wazuh env[1475]: 2025/10/03 02:50:30 wazuh-modulesd:content_manager: INFO: Loaded content_manager module.
Oct 03 02:50:31 Wazuh env[832]: Started wazuh-modulesd...
Oct 03 02:50:33 Wazuh env[832]: Completed.
Oct 03 02:50:33 Wazuh systemd[1]: Started wazuh-manager.service - Wazuh manager.
root@Wazuh:~# 

```

Access the Wazuh Web Interface

1. Open your browser and go to: <https://{your-wazuh-vm-ip}>
2. If you cannot access it, allow HTTPS traffic (port 443): `sudo ufw allow 443`
3. Try again — you should now see the Wazuh dashboard.
4. Log in using the **username** and **password** you saved earlier.

Once logged in, you should see the Wazuh Dashboard Overview showing system status, agent summary, and alert statistics. This confirms that your Wazuh installation is active and operational. 



👉 If it still doesn't work, check your **cloud provider firewall settings** (DigitalOcean, AWS, Azure, Vultr, etc.) and allow port **443**.

Installing TheHive

Register for TheHive

Go to the official StrangeBee portal and register for an account: 👉 [Register here](#)

📍 **Note:** TheHive requires a work or student email. They also provide a free trial.

Connect to TheHive VM

- Open PowerShell on your local machine and connect via SSH (replace with your TheHive VM IP): `ssh root@{your-thehive-vm-ip}`
- Type **yes** when prompted, then enter your VM password.
- Update your system packages: `sudo apt update && apt upgrade -y`

Install Dependencies

Follow the official StrangeBee installation guide:

👉 [TheHive Installation Guide \(Linux Standalone Server\)](#)

Copy and run the commands exactly as shown in the documentation.



TheHive

Overview

Installation

Installation Methods

System Requirements

Single Server Installation

Install on Linux

Step 1: Install required dependencies

Step 2: Set up the Java Virtual Machine

Step 3: Install and configure Apache Cassandra

Step 4: Install and configure Elasticsearch

Step 5: Install and configure TheHive

Advanced configuration

Backup

Monitoring

One-Command Install

Deploy with Docker

Cluster Deployment

Home

TheHive

Cortex

Cortex Neurons

Resources

Search

Step 1: Install required dependencies

Start by installing the necessary dependencies for TheHive.

DEB (Debian/Ubuntu) **RPM (RHEL/Fedora)**

Run the following commands:

```
sudo apt update  
sudo apt install wget curl gnupg coreutils apt-transport-https git ca-certificates ca-certificates-java software-properties-common
```

[Copy to clipboard](#)

Step 2: Set up the Java Virtual Machine

TheHive requires Java to run its application server and to manage various processes.

Java support

- For security and long-term support, we recommend using [Amazon Corretto](#), which provides OpenJDK builds maintained by Amazon.
- Only Java 11 is supported. Java 8 is no longer supported.

DEB RPM Other installation methods

```
root@The-Hive:~# sudo apt update
sudo apt install wget curl gnupg coreutils apt-transport-https git ca-certificates ca-certificates-java software-properties-common python3-pip l
sb-release unzip
Hit:1 https://repos-droplet.digitalocean.com/apt/droplet-agent main InRelease
Hit:2 http://mirrors.digitalocean.com/ubuntu noble InRelease
Hit:3 http://mirrors.digitalocean.com/ubuntu noble-updates InRelease
Hit:4 http://mirrors.digitalocean.com/ubuntu noble-backports InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
3 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
wget is already the newest version (1.21.4-1ubuntu4.1).
wget set to manually installed.
curl is already the newest version (8.5.0-2ubuntu10.6).
curl set to manually installed.
gnupg is already the newest version (2.4.4-2ubuntu17.3).
gnupg set to manually installed.
coreutils is already the newest version (9.4-3ubuntu6.1).
coreutils set to manually installed.
apt-transport-https is already the newest version (2.8.3).
git is already the newest version (1:2.43.0-1ubuntu7.3).
git set to manually installed.
ca-certificates is already the newest version (20240203).
software-properties-common is already the newest version (0.99.49.3).
software-properties-common set to manually installed.
lsb-release is already the newest version (12.0-2).
lsb-release set to manually installed.
The following additional packages will be installed:
  binutils binutils-common binutils-x86_64-linux-gnu build-essential bzip2 cpp cpp-13 cpp-13-x86_64-linux-gnu cpp-x86_64-linux-gnu dpkg-dev
  fakeroot g++-13 g++-13-x86_64-linux-gnu g++-x86_64-linux-gnu gcc gcc-13 gcc-13-base gcc-13-x86_64-linux-gnu gcc-x86_64-linux-gnu
  javascript-common libalgoritm-diff-perl libalgoritm-diff-xxs-perl libalgoritm-merge-perl libbasan8 libatomic1 libbinutils libccl1-0
  libctf-nobfd0 libctf0 libdpkg-perl libexpat1-dev libfakeroot libfile-fcntllock-perl libgcc-13-dev libgomp1 libgprofng0 libhwasan0 libisl123
  libitm1 libjs-jquery libjs-sphinxdoc libjs-underscore liblsan0 libmpc3 libpython3-3-dev libpython3.12-dev libquadmath0 libsframe1
  libstdc++-13-dev libtsan2 libubsan1 lto-disabled-list make python3-dev python3-wheel python3.12-dev zlib1g-dev
Suggested packages:
  binutils-doc gprofng-gui bzip2-doc cpp-doc gcc-13-locales cpp-13-doc debian-keyring g++-multilib g++-13-multilib gcc-13-doc gcc-multilib
```

TheHive
Overview
Installation
Installation Methods
System Requirements
Single Server Installation
Install on Linux
Step 1: Install required dependencies
Step 2: Set up the Java Virtual Machine
Step 3: Install and configure Apache Cassandra
Step 4: Install and configure Elasticsearch
Step 5: Install and configure TheHive
Advanced configuration
Backup
Monitoring
One-Command Install
Deploy with Docker
Cluster Deployment
Licenses
Version Upgrades
Configuration
Operations
Administration Guides
User Guides

Step 2: Set up the Java Virtual Machine

TheHive requires Java to run its application server and to manage various processes.

Java support

- For security and long-term support, we recommend using [Amazon Corretto](#), which provides OpenJDK builds maintained by Amazon.
- Only Java 11 is supported. Java 8 is no longer supported.

DEB RPM Other installation methods

1. Run the following commands:

```
wget -qO- https://apt.corretto.aws/corretto.key | sudo gpg --dearmor -o /usr/share/keyrings/corretto.gpg
echo "deb [signed-by=/usr/share/keyrings/corretto.gpg] https://apt.corretto.aws stable main" | sudo tee -a /etc/apt/sources.list.d/corretto.sources.list
sudo apt update
sudo apt install java-common java-11-amazon-corretto-jdk
echo JAVA_HOME="/usr/lib/jvm/java-11-amazon-corretto" | sudo tee -a /etc/environment
export JAVA_HOME="/usr/lib/jvm/java-11-amazon-corretto"
deb [signed-by=/usr/share/keyrings/corretto.gpg] https://apt.corretto.aws stable main
Hit:1 https://repos-droplet.digitalocean.com/apt/droplet-agent main InRelease
Get:2 https://apt.corretto.aws stable InRelease [10.7 kB]
Hit:3 http://mirrors.digitalocean.com/ubuntu noble InRelease
Get:4 http://mirrors.digitalocean.com/ubuntu noble-updates InRelease [126 kB]
Get:5 https://apt.corretto.aws stable/main amd64 Packages [22.5 kB]
Hit:6 http://mirrors.digitalocean.com/ubuntu noble-backports InRelease
Get:7 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
0% [Waiting for headers] [7 InRelease 14.2 kB/126 kB 11%]
```

2. Verify the installation.

```
java -version
```

You should see output similar to the following:

```
openjdk version "11.0.28" 2025-07-15
OpenJDK Runtime Environment Corretto-11.0.28.6.1 (build 11.0.28+6-LTS)
OpenJDK 64-Bit Server VM Corretto-11.0.28.6.1 (build 11.0.28+6-LTS, mixed mode)
```

If a different Java version appears, set Java 11 as the default using `sudo update-alternatives --config java`.

```
root@The-Hive:~# wget -qO- https://apt.corretto.aws/corretto.key | sudo gpg --dearmor -o /usr/share/keyrings/corretto.gpg
echo "deb [signed-by=/usr/share/keyrings/corretto.gpg] https://apt.corretto.aws stable main" | sudo tee -a /etc/apt/sources.list.d/corretto.sources.list
sudo apt update
sudo apt install java-common java-11-amazon-corretto-jdk
echo JAVA_HOME="/usr/lib/jvm/java-11-amazon-corretto" | sudo tee -a /etc/environment
export JAVA_HOME="/usr/lib/jvm/java-11-amazon-corretto"
deb [signed-by=/usr/share/keyrings/corretto.gpg] https://apt.corretto.aws stable main
Hit:1 https://repos-droplet.digitalocean.com/apt/droplet-agent main InRelease
Get:2 https://apt.corretto.aws stable InRelease [10.7 kB]
Hit:3 http://mirrors.digitalocean.com/ubuntu noble InRelease
Get:4 http://mirrors.digitalocean.com/ubuntu noble-updates InRelease [126 kB]
Get:5 https://apt.corretto.aws stable/main amd64 Packages [22.5 kB]
Hit:6 http://mirrors.digitalocean.com/ubuntu noble-backports InRelease
Get:7 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
0% [Waiting for headers] [7 InRelease 14.2 kB/126 kB 11%]
```

After installing Java, verify the version: `java -version`

If no version is displayed, rerun the Java installation command (the one starting with `wget ...`),

then confirm installation again with: `java -version`

Install Cassandra

Follow the official Apache Cassandra installation guide

TheHive
Overview
Installation
Installation Methods
System Requirements
Single Server Installation
Install on Linux
Step 1: Install required dependencies
Step 2: Set up the Java virtual machine (JVM)
Step 3: Install and configure Apache Cassandra
Step 4: Install and configure Elasticsearch
Step 5: Install and configure TheHive
Advanced configuration
Backup
Monitoring
One-Command Install
Deploy with Docker
Cluster Deployment
Licenses
Version Upgrades
Configuration
Operations
Administration Guides
User Guides
API
Release Notes

Step 3.1: Install Cassandra

DEB RPM Other installation methods

- Add Cassandra repository references.
 - Download Cassandra repository keys.

```
wget -qO - https://downloads.apache.org/cassandra/KEYS | sudo gpg --dearmor -o /usr/share/keyrings/cassandra-archive.gpg
```
- Check if the /etc/apt/sources.list.d/cassandra.sources.list file exists. If it doesn't, create it.
- Add the repository to your system by appending the following line to the /etc/apt/sources.list.d/cassandra.sources.list file.

```
echo "deb [signed-by=/usr/share/keyrings/cassandra-archive.gpg] https://debian.cassandra.apache.org 41x main"
```

- Update your package index and install Cassandra using the following commands:

```
sudo apt update  
sudo apt install cassandra
```

- Verify that Cassandra is installed.

```
systemctl status cassandra
```

This will show you the status of the Cassandra service. If Cassandra is installed you should see details about the service. If it's not installed, you will receive an error indicating that the service isn't found.

- By default, data is stored in /var/lib/cassandra. Set appropriate permissions for this directory to avoid any issues with data storage and access.

```
sudo chown -R cassandra:cassandra /var/lib/cassandra
```

Refer to the official Cassandra documentation website for the most up-to-date instructions.

```
root@The-Hive:~# wget -qO - https://downloads.apache.org/cassandra/KEYS | sudo gpg --dearmor -o /usr/share/keyrings/cassandra-archive.gpg  
root@The-Hive:~# echo "deb [signed-by=/usr/share/keyrings/cassandra-archive.gpg] https://debian.cassandra.apache.org 41x main" | sudo tee -a /etc/apt/sources.list.d/cassandra.sources.list  
root@The-Hive:~# sudo apt update  
root@The-Hive:~# sudo apt upgrade  
Hit:1 https://apt.corretto.aws stable InRelease  
Hit:2 https://repos-droplet.digitalocean.com/apt/droplet-agent main InRelease  
Hit:3 http://mirrors.digitalocean.com/ubuntu noble InRelease  
Hit:4 http://mirrors.digitalocean.com/ubuntu noble-updates InRelease  
Hit:5 http://mirrors.digitalocean.com/ubuntu noble-backports InRelease  
Get:6 https://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]  
Get:7 https://thehive.jfrog.io/artifactory/cassandra-deb 41x/main amd64 Packages [3902 B]  
Get:8 https://thehive.jfrog.io/artifactory/cassandra-deb 41x/main amd64 Packages [696 B]  
Fetched 131 kB in 3s (56.1 kB/s)  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
3 packages can be upgraded. Run 'apt list --upgradable' to see them.  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
Suggested packages:  
  cassandra-tools  
The following NEW packages will be installed:  
  cassandra  
0 upgraded, 1 newly installed, 0 to remove and 3 not upgraded.  
Need to get 50.7 MB of archives.  
After this operation, 61.3 MB of additional disk space will be used.  
Get:1 https://apt.corretto.aws stable InRelease [50.7 MB]  
Fetched 50.7 MB in 8s (6326 kB/s)  
Selecting previously unselected package cassandra.  
(Reading database ... 119329 files and directories currently installed.)  
Preparing to unpack .../cassandra_4.1.10_all.deb ...  
Unpacking cassandra (4.1.10) ...  
Setting up cassandra (4.1.10) ...  
info: Selecting GID '100' range '100 to 999' ...  
info: Selecting group 'cassandra' (GID 112) ...  
vm.max_map_count=1048575  
net.ipv4.tcp_keepalive_time=300  
update-rc.d: warning: start and stop actions are no longer supported; falling back to defaults  
Scanning processes...  
Scanning linux images...  
Running kernel seems to be up-to-date.  
No services need to be restarted.  
No containers need to be restarted.  
No user sessions are running outdated binaries.  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
root@The-Hive:~# systemctl status cassandra  
● cassandra.service - LSB: distributed storage system for structured data  
  Loaded: loaded (/etc/init.d/cassandra; generated)  
  Active: active (running) since Fri 2025-09-12 09:28:58 UTC; 9s ago  
    Docs: man:systemd-sysv-generator(8)  
  Process: 5888 ExecStart=/etc/init.d/cassandra start (code=exited, status=0/SUCCESS)  
    Tasks: 26 (limit: 9486)  
   Memory: 2.1G (peak: 2.1G)  
    CPU: 13.125s  
     CGroup: /system.slice/cassandra.service  
             └─58946 /usr/bin/java -ea -da:net.openhft... -XX:+UseThreadPriorities -XX:+HeapDumpOnOutOfMemoryError -Xss256k -XX:+AlwaysPreTouch  
  
Sep 12 09:28:58 The-Hive systemd[1]: Starting cassandra.service - LSB: distributed storage system for structured data...  
Sep 12 09:28:58 The-Hive systemd[1]: Started cassandra.service - LSB: distributed storage system for structured data.  
root@The-Hive:~# sudo chown -R cassandra:cassandra /var/lib/cassandra
```

Once installed, configure it properly before starting TheHive.

Install Elasticsearch

Next, install Elasticsearch.

The screenshot shows a web-based documentation interface for Elasticsearch installation. On the left, a sidebar lists various installation paths: Overview, Installation (selected), Installation Methods, System Requirements, Single Server Installation, Install on Linux (selected), Step 1: Install required dependencies, Step 2: Set up the Java virtual machine (JVM), Step 3: Install and configure Apache Cassandra, Step 4: Install and configure Elasticsearch, Step 5: Install and configure TheHive, Advanced configuration, Backup, Monitoring, One-Command Install, Deploy with Docker, Cluster Deployment, Licenses, and Version Upgrades. The main content area is titled "Step 4.1: Install Elasticsearch". It shows three tabs: DEB (selected), RPM, and Other installation methods. Under the DEB tab, step 1 is described as "Add Elasticsearch repository references." It includes two command examples: "wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo gpg --dearmor -o /usr/share/keyrings/elasticsearch-keyring.gpg" and "echo \"deb [signed-by=/usr/share/keyrings/elasticsearch-keyring.gpg] https://artifacts.elastic.co/packages/7.x/stable main\" | sudo tee /etc/apt/sources.list.d/elasticsearch-7.x.list". Step 2 is described as "Update your package index and install Elasticsearch using the following commands:" with the command "sudo apt update" followed by "sudo apt install elasticsearch". A note at the bottom says "Refer to the official Elasticsearch documentation website for the most up-to-date instructions."

```
root@The-Hive:~#
root@The-Hive:~# wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo gpg --dearmor -o /usr/share/keyrings/elasticsearch-keyring.gpg
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apt-transport-https is already the newest version (2.8.3).
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
root@The-Hive:~# curl -s https://artifacts.elastic.co/packages/7.x/stable/main | sudo tee /etc/apt/sources.list.d/elasticsearch-7.x.list
deb [signed-by=/usr/share/keyrings/elasticsearch-keyring.gpg] https://artifacts.elastic.co/packages/7.x/stable main
root@The-Hive:~# sudo apt update
sudo apt install elasticsearch
Hit:1 https://repository.digitalocean.com/ubuntu/droplet-agent main InRelease
Hit:2 https://repository.digitalocean.com/ubuntu stable InRelease
Get:3 https://artifacts.elastic.co/packages/7.x/stable InRelease [13.7 kB]
Hit:4 http://mirrors.digitalocean.com/ubuntu_noble InRelease
Hit:5 http://mirrors.digitalocean.com/ubuntu_noble-updates InRelease
Get:6 https://artifacts.elastic.co/packages/7.x/stable/main amd64 Packages [145 kB]
Hit:7 http://mirrors.digitalocean.com/ubuntu_noble-backports InRelease
Hit:8 http://security.ubuntu.com/ubuntu_noble-security InRelease
Hit:9 https://apache.jfrog.io/artifactory/cassandra-deb 41x InRelease
Fetched 158 kB in 9s (18.5 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
3 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  elasticsearch
0 upgraded, 1 newly installed, 0 to remove and 3 not upgraded.
Need to get 325 MB of archives.
After this operation, 542 MB of additional disk space will be used.
Get:1 https://artifacts.elastic.co/packages/7.x/stable/main amd64 elasticsearch amd64 7.17.29 [325 MB]
Fetched 325 MB in 5s (66.9 MB/s)
Selecting previously unselected package elasticsearch.
(Reading database ... 110493 files and directories currently installed.)
Preparing to unpack .../elasticsearch_7.17.29_amd64.deb ...
Creating elasticsearch group... OK
Creating elasticsearch user... OK
Unpacking elasticsearch (7.17.29) ...
Setting up elasticsearch (7.17.29) ...
### NOT starting on installation, please execute the following statements to configure elasticsearch service to start automatically using systemd
sudo systemctl daemon-reload
sudo systemctl enable elasticsearch.service
## You can start elasticsearch service by executing
sudo systemctl start elasticsearch.service
warning: usage of JAVA_HOME is deprecated, use ES_JAVA_HOME
Created elasticsearch keystore in /etc/elasticsearch/elasticsearch.keystore
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

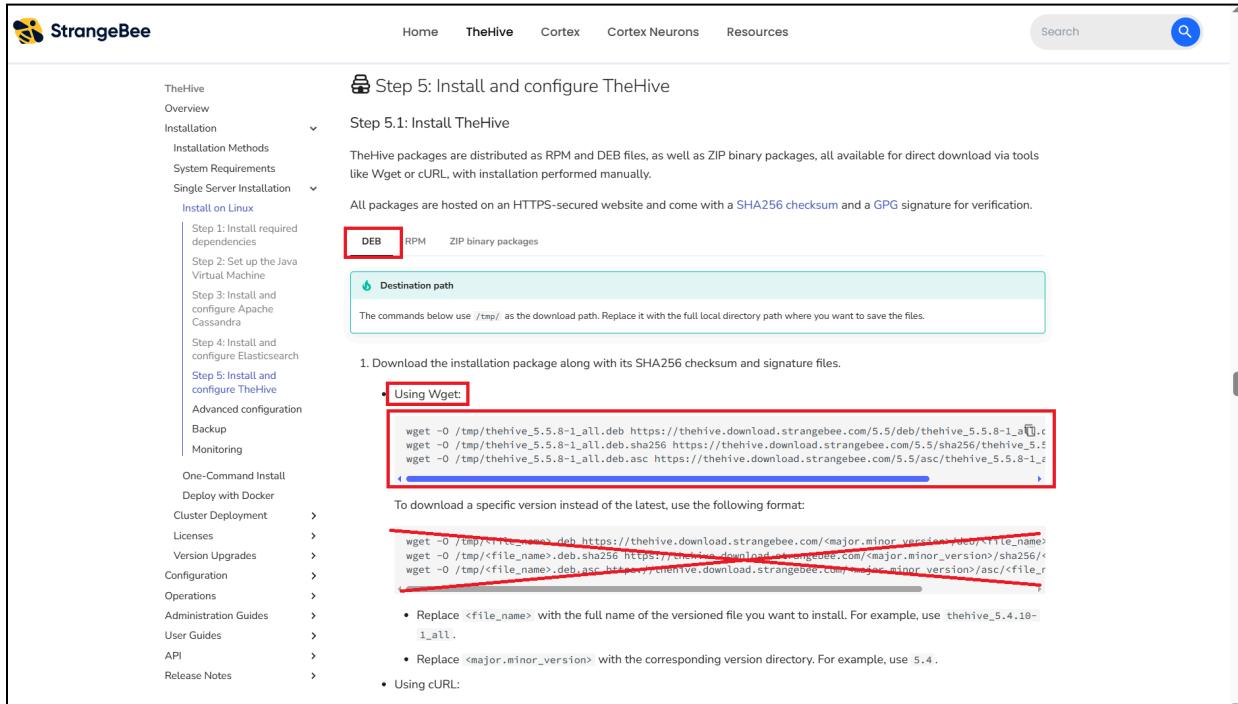
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@The-Hive:~|
```

Install TheHive

Use **wget** (recommended) to download and install TheHive.

👉 If you prefer **curl** or another method, scroll down in the StrangeBee docs and copy the relevant command:



The screenshot shows the StrangeBee documentation for TheHive. The left sidebar has a tree view of installation methods: Overview, Installation Methods, System Requirements, Single Server Installation, and Install on Linux. Under Install on Linux, Step 1: Install required dependencies is selected. The main content area is titled "Step 5: Install and configure TheHive". It has a sub-section "Step 5.1: Install TheHive". Below that is a note about RPM and ZIP binary packages, with "DEB" highlighted with a red box. A "Destination path" section follows, containing a code block for wget. A red box highlights the "Using Wget" section. Below it is a note about downloading specific versions. At the bottom, there's a "Using cURL:" section. The terminal window at the bottom shows the execution of the wget command to download the deb file, its sha256 checksum, and its asc file.

```
root@The-Hive:~# wget -O /tmp/thehive_5.5.8-1_all.deb https://thehive.download.strangebee.com/5.5/deb/thehive_5.5.8-1_all.deb
wget -O /tmp/thehive_5.5.8-1_all.deb.sha256 https://thehive.download.strangebee.com/5.5/sha256/thehive_5.5.8-1_all.deb.sha256
--2025-09-12 09:47:32-- https://thehive.download.strangebee.com/5.5/deb/thehive_5.5.8-1_all.deb
Resolving thehive.download.strangebee.com (thehive.download.strangebee.com)... 3.164.68.11, 3.164.68.103, 3.164.68.79, ...
Connecting to thehive.download.strangebee.com (thehive.download.strangebee.com)|3.164.68.11|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 472391872 (451M) [application/vnd.debian.binary-package]
Saving to: '/tmp/thehive_5.5.8-1_all.deb'

/tmpp/thehive_5.5.8-1_all.deb      100%[=====] 450.51M 6.88MB/s   in 82s

2025-09-12 09:48:56 (5.51 MB/s) - '/tmp/thehive_5.5.8-1_all.deb' saved [472391872/472391872]

--2025-09-12 09:48:56-- https://thehive.download.strangebee.com/5.5/sha256/thehive_5.5.8-1_all.deb.sha256
Resolving thehive.download.strangebee.com (thehive.download.strangebee.com)... 3.164.68.11, 3.164.68.124, 3.164.68.103, ...
Connecting to thehive.download.strangebee.com (thehive.download.strangebee.com)|3.164.68.11|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 98 [binary/octet-stream]
Saving to: '/tmp/thehive_5.5.8-1_all.deb.sha256'

/tmpp/thehive_5.5.8-1_all.deb.sha256    100%[=====] 90 --.-KB/s   in 0s

2025-09-12 09:48:57 (49.2 MB/s) - '/tmp/thehive_5.5.8-1_all.deb.sha256' saved [90/90]

--2025-09-12 09:48:57-- https://thehive.download.strangebee.com/5.5/asc/thehive_5.5.8-1_all.deb.asc
Resolving thehive.download.strangebee.com (thehive.download.strangebee.com)... 3.164.68.103, 3.164.68.124, 3.164.68.11, ...
Connecting to thehive.download.strangebee.com (thehive.download.strangebee.com)|3.164.68.103|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 833 [application/ppk-keys]
Saving to: '/tmp/thehive_5.5.8-1_all.deb.asc'

/tmpp/thehive_5.5.8-1_all.deb.asc      100%[=====] 833 --.-KB/s   in 0s

2025-09-12 09:48:58 (38.2 MB/s) - '/tmp/thehive_5.5.8-1_all.deb.asc' saved [833/833]

root@The-Hive:~# |
```

TheHive

Overview

Installation

- Installation Methods
- System Requirements
- Single Server Installation
 - Install on Linux
 - Step 1: Install required dependencies
 - Step 2: Set up the Java Virtual Machine
 - Step 3: Install and configure Apache Cassandra
 - Step 4: Install and configure Elasticsearch
 - Step 5: Install and configure TheHive**
 - Advanced configuration
 - Backup
 - Monitoring
 - One-Command Install
 - Deploy with Docker
- Cluster Deployment
- Licenses
- Version Upgrades
- Configuration
- Operations
- Administration Guides
- User Guides
- API
- Release Notes

Home TheHive Cortex Cortex Neurons Resources Search

Expected GPG warning

gpg: WARNING: This key is not certified with a trusted signature!
gpg: There is no indication that the signature belongs to the owner.

This warning is expected. It means the package is signed with the official TheHive release key, but you haven't marked this key as `trusted` in your local GPG setup. As long as you see `Good` signature and the fingerprint matches, the verification is successful. Don't mark our key as globally trusted—the warning is a normal safety reminder and should remain visible.

If you don't see `Good` signature, if the fingerprint differs, or if the signature is reported as `BAD`, don't install the package. This indicates the integrity or authenticity of the file can't be confirmed. Report the issue to the StrangeBee Security Team.

3. Install the package.

- Using `apt-get` to manage dependencies automatically:

```
sudo apt-get install /tmp/thehive_5.5.8-1_all.deb
```

- Using `dpkg`:

```
sudo dpkg -i /tmp/thehive_5.5.8-1_all.deb
```

Step 5.2: Configure TheHive

Standalone server configuration

In this guide, we will configure TheHive for a standalone server, with all components hosted on a single server. This setup is suitable for testing and production environments. For cluster deployments, refer to [Setting up a Cluster with TheHive](#).

- Open the `/etc/thehive/application.conf` file using a text editor.
- In the `application.conf` file, configure the service.

To ensure TheHive works correctly—especially with authentication mechanisms such as Single Sign-On (SSO)—configure the

```
root@The-Hive:~# sudo apt-get install /tmp/thehive_5.5.8-1_all.deb
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'thehive' instead of '/tmp/thehive_5.5.8-1_all.deb'
The following NEW packages will be installed:
  thehive
0 upgraded, 1 newly installed, 0 to remove and 3 not upgraded.
Need to get 0 B/472 MB of archives.
After this operation, 472 MB of additional disk space will be used.
Get:1 /tmp/thehive_5.5.8-1_all.deb thehive all 5.5.8-1 [472 MB]
Selecting previously unselected package thehive.
(Reading database ... 111591 files and directories currently installed.)
Preparing to unpack /tmp/thehive_5.5.8-1_all.deb ...
Unpacking thehive (5.5.8-1) ...
Setting up thehive (5.5.8-1) ...
Creating system group: thehive
Creating system user: thehive in thehive with thehive daemon-user and shell /bin/false
1+0 records in
1+0 records out
1024 bytes (1.0 kB, 1.0 KiB) copied, 5.9664e-05 s, 17.2 MB/s
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@The-Hive:~#
```

Configure Cassandra

Open the Cassandra configuration file: `nano /etc/cassandra/cassandra.yaml`

If `cassandra.yaml` is missing, Cassandra likely didn't install correctly. Reinstall it before proceeding.

Edit the following keys (use **CTRL+W** in nano to search each key):

```
cluster_name: 'soc-workflow'  
listen_address: {thehive-vm-ip}  
rpc_address: {thehive-vm-ip}  
- seeds: "{thehive-vm-ip}:7000"
```

Save and exit: **CTRL+X** → **Y** → **Enter**

Restart Cassandra cleanly:

```
sudo systemctl stop cassandra.service  
sudo rm -rf /var/lib/cassandra/*  
sudo systemctl start cassandra.service  
sudo systemctl status cassandra.service
```

✓ If the status shows **active (running)**, Cassandra is ready.

```
root@The-Hive:~# systemctl stop cassandra.service  
root@The-Hive:~# rm -rf /var/lib/cassandra/*  
root@The-Hive:~# systemctl start cassandra.service  
root@The-Hive:~# systemctl status cassandra.service  
● cassandra.service - LSB: distributed storage system for structured data  
  Loaded: loaded (/etc/init.d/cassandra; generated)  
  Active: active (running) since Fri 2025-09-12 10:26:34 UTC; 9s ago  
    Docs: man:systemd-sysv-generator(8)  
  Process: 7637 ExecStart=/etc/init.d/cassandra start (code=exited, status=0/SUCCESS)  
  Tasks: 46 (limit: 9486)  
  Memory: 2.2G (peak: 2.2G)  
  CPU: 19.533s  
  CGroup: /system.slice/cassandra.service  
          └─7741 /usr/bin/java -ea -da:net.openhft... -XX:+UseThreadPriorities -XX:+HeapDumpOnOutOfMemoryError -Xss256k -XX:+AlwaysPreTouch -XX:-UseBiasP  
  
Sep 12 10:26:34 The-Hive systemd[1]: Starting cassandra.service - LSB: distributed storage system for structured data...  
Sep 12 10:26:34 The-Hive systemd[1]: Started cassandra.service - LSB: distributed storage system for structured data.  
root@The-Hive:~# |
```

Configure Elasticsearch

Open the configuration file: `nano /etc/elasticsearch/elasticsearch.yml`

⚠️ If `elasticsearch.yml` is missing, Elasticsearch likely didn't install correctly. Reinstall it using the previous steps.

Then open the configuration file again: `nano /etc/elasticsearch/elasticsearch.yml`

Update the following keys (use **CTRL+W** in nano to search each key):

```
cluster.name: soc-workflow
```

```

node.name: node-1

network.host: {hive-vm-ip}

http.port: 9200

cluster.initial_master_nodes: ["node-1"]

```

```

GNU nano 7.2
# Please consult the documentation for further information on configuration options:
# https://www.elastic.co/guide/en/elasticsearch/reference/index.html
#
# Cluster
#
# Use a descriptive name for your cluster:
#
#cluster.name: soc-workflow
#
# Node
#
# Use a descriptive name for the node:
#
node.name: node-1
#
# Add custom attributes to the node:
#
#node.attr.rack: r1
#
# Paths
#
# Path to directory where to store the data (separate multiple locations by comma):
#
path.data: /var/lib/elasticsearch
#
# Path to log files:
#
path.logs: /var/log/elasticsearch
#
# Memory
#
# Lock the memory on startup:
#
#bootstrap.memory_lock: true
#
# Make sure that the heap size is set to about half the memory available
# on the system and that the owner of the process is allowed to use this
# limit.
#
# Elasticsearch performs poorly when the system is swapping the memory.
#
# Network
#
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
#
network.host: 68.183.84.53
#
# By default Elasticsearch listens for HTTP traffic on the first free port it
# finds starting at 9200. Set a specific HTTP port here:
#
http.port: 9200
#
# For more information, consult the network module documentation.
#
# Discovery
#
# Pass an initial list of hosts to perform discovery when this node is started:
# The default list of hosts is ["127.0.0.1", "[::1]"]
#
discovery.seed_hosts: ["host1", "host2"]
#
# Bootstrap the cluster using an initial set of master-eligible nodes:
#
cluster.initial_master_nodes: ["node-1"]
#
# For more information, consult the discovery and cluster formation module documentation.
#
# Various
#
# Require explicit names when deleting indices:

```

Save and exit: **CTRL+X → Y → Enter**

Restart and enable Elasticsearch:

```
sudo systemctl start elasticsearch.service
```

```
sudo systemctl enable elasticsearch.service  
sudo systemctl status elasticsearch.service
```

- ✓ If the status shows **active (running)**, Elasticsearch is ready.

Configure TheHive

Before configuring TheHive, update file permissions for its directory:

```
cd /opt/thp  
ls -l  
sudo chown -R thehive:thehive /opt/thp  
ls -l
```

- This ensures TheHive has proper access to its files.

```
root@The-Hive:~# cd /opt/thp
root@The-Hive:/opt/thp# ll
total 12
drwxr-xr-x 3 root root 4096 Sep 12 09:50 .
drwxr-xr-x 5 root root 4096 Sep 12 09:50 ../
drwxr-xr-x 5 root root 4096 Sep 12 09:50 thehive/
root@The-Hive:/opt/thp# chown -R thehive:thehive /opt/thp
root@The-Hive:/opt/thp# ll
total 12
drwxr-xr-x 3 thehive thehive 4096 Sep 12 09:50 .
drwxr-xr-x 5 root      root    4096 Sep 12 09:50 ../
drwxr-xr-x 5 thehive thehive 4096 Sep 12 09:50 thehive/
root@The-Hive:/opt/thp#
```

Open the configuration file: `sudo nano /etc/thehive/application.conf`

 If `application.conf` is missing, TheHive likely didn't install correctly. Reinstall it before proceeding.

Edit the following keys (use **CTRL+W** in nano to search each one):

```

hostname = ["{thehive-vm-ip}"]

cluster-name = soc-workflow

backend = elasticsearch
    hostname = ["{thehive-vm-ip}"]

application.baseUrl = "http://{thehive-vm-ip}:9000"

```

```

GNU nano 7.2                               /etc/thehive/application.conf

# Database and index configuration
# By default, TheHive is configured to connect to local Cassandra 4.x and a
# local Elasticsearch services without authentication.
db.janusgraph {
storage {
    backend = cql
    hostname = ["68.183.84.53"]
    # Cassandra authentication (if configured)
    # username = "thehive"
    # password = "password"
    cql {
        cluster-name = soc-workflow
        keyspace = thehive
    }
}
index.search {
    backend = elasticsearch
    hostname = ["68.183.84.53"]
    index-name = thehive
}
}

# Attachment storage configuration
# By default, TheHive is configured to store files locally in the folder.
# The path can be updated and should belong to the user/group running thehive service. (by default: thehive:thehive)
storage {
provider = localfs
localfs.location = /opt/thp/thehive/files
}

# Define the maximum size for an attachment accepted by TheHive
play.http.parser.maxDiskBuffer = 1GB
# Define maximum size of http request (except attachment)
play.http.parser.maxMemoryBuffer = 10M

# Service configuration
application.baseUrl = "http://68.183.84.53:9000"
play.http.context = "/"

```

File menu: G Help, X Exit, W Write Out, R Read File, N Where Is, U Replace, K Cut, J Paste, T Execute, J Justify, C Location, / Go To Line, U Undo, E Redo, A Set Mark, M-Copy

Save and exit: **CTRL+X → Y → Enter**

Restart and enable TheHive

```

sudo systemctl start thehive.service
sudo systemctl enable thehive.service
sudo systemctl status thehive.service

```

✓ If the status shows **active (running)**, TheHive is ready.

```
root@The-Hive:/opt/thp# systemctl start thehive
root@The-Hive:/opt/thp# systemctl enable thehive
Created symlink /etc/systemd/system/multi-user.target.wants/thehive.service → /usr/lib/systemd/system/thehive.service.
root@The-Hive:/opt/thp# systemctl status thehive
● thehive.service - Scalable, Open Source and Free Security Incident Response Solutions
  Loaded: loaded (/usr/lib/systemd/system/thehive.service; enabled; preset: enabled)
  Active: active (running) since Fri 2025-09-12 10:47:50 UTC; 17s ago
    Docs: https://strangebee.com
    Main PID: 8411 (java)
       Tasks: 53 (limit: 9486)
      Memory: 509.5M (peak: 509.5M)
        CPU: 34.822s
       CGroup: /system.slice/thehive.service
               └─8411 java -Dfile.encoding=UTF-8 -Dconfig.file=/etc/thehive/application.conf -Dlogger.file=/etc/thehive/logback.xml -Dlogback.configurationFile=/etc/thehive/logback.xml

Sep 12 10:47:50 The-Hive systemd[1]: Started thehive.service - Scalable, Open Source and Free Security Incident Response Solutions.
root@The-Hive:/opt/thp#
```

Verify all required services are running

```
sudo systemctl status cassandra.service
sudo systemctl status elasticsearch.service
sudo systemctl status thehive.service
```

All three must show **active (running)** before you proceed.

```
root@The-Hive:~# systemctl status elasticsearch.service
● elasticsearch.service - Elasticsearch
  Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; enabled; preset: enabled)
  Active: active (running) since Fri 2025-10-03 05:33:56 UTC; 24h ago
    Docs: https://www.elastic.co
    Main PID: 9305 (java)
       Tasks: 79 (limit: 9486)
      Memory: 3.3G (peak: 3.5G)
        CPU: 5min 34.828s
       CGroup: /system.slice/elasticsearch.service
               └─9305 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des.networkaddress.cache.ttl=60 -Des.networkaddress.cache.negative.ttl=10 -XX:+AlwaysPreTouch -Xss1m -Djava.awt.headless=true
                 ├─9525 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/bin/controller

Oct 03 05:33:29 The-Hive systemd[1]: Starting elasticsearch.service - Elasticsearch...
Oct 03 05:33:35 The-Hive systemd-entropy[9305]: Oct 03 2025 5:33:35 AM sun.util.locale.provider.LocaleProviderAdapter <clinit>
Oct 03 05:33:35 The-Hive systemd-entropy[9305]: WARNING: COMPAT locale provider will be removed in a future release
Oct 03 05:33:56 The-Hive systemd[1]: Started elasticsearch.service - Elasticsearch.

root@The-Hive:~# systemctl status cassandra.service
● cassandra.service - LSB: distributed storage system for structured data
  Loaded: loaded (/etc/init.d/cassandra; generated)
  Active: active (running) since Fri 2025-10-03 05:33:36 UTC; 24h ago
    Docs: man:systemd-sysv-generator(8)
  Process: 9532 ExecStart=/etc/init.d/cassandra start (code=exited, status=0/SUCCESS)
    Tasks: 82 (limit: 9486)
   Memory: 2.3G (peak: 2.3G)
     CPU: 30min 26.481s
    CGroup: /system.slice/cassandra.service
            └─9636 /usr/bin/java -ea -da:net.openhft... -XX:+UseThreadPriorities -XX:+HeapDumpOnOutOfMemoryError -Xss256k -XX:+AlwaysPreTouch -XX:-UseBiasedLocking -XX:+UseTLAB -XX:+Re...

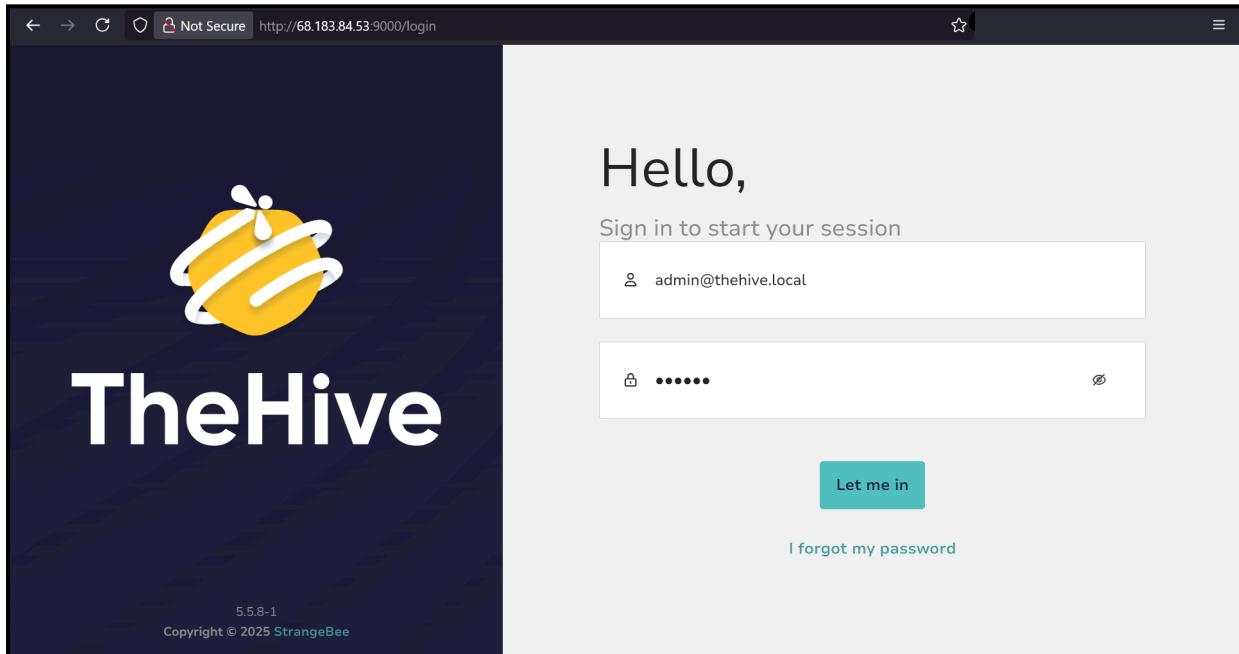
Oct 03 05:33:36 The-Hive systemd[1]: Starting cassandra.service - LSB: distributed storage system for structured data...
Oct 03 05:33:36 The-Hive systemd[1]: Started cassandra.service - LSB: distributed storage system for structured data.
root@The-Hive:~# systemctl status thehive.service
● thehive.service - Scalable, Open Source and Free Security Incident Response Solutions
  Loaded: loaded (/usr/lib/systemd/system/thehive.service; enabled; preset: enabled)
  Active: active (running) since Fri 2025-10-03 05:34:22 UTC; 24h ago
    Docs: https://strangebee.com
    Main PID: 9815 (java)
       Tasks: 115 (limit: 9486)
      Memory: 963.2M (peak: 1.0G)
        CPU: 1h 2min 37.019s
       CGroup: /system.slice/thehive.service
               └─9815 java -Dfile.encoding=UTF-8 -Dconfig.file=/etc/thehive/application.conf -Dlogger.file=/etc/thehive/logback.xml -Dpidfile.path=/dev/null -cp /opt/thehive/lib/org.thp.i

Oct 03 05:34:22 The-Hive systemd[1]: Started thehive.service - Scalable, Open Source and Free Security Incident Response Solutions.
root@The-Hive:~#
```

Once all services are running, open your browser and go to:

<http://{your-hive-vm-ip}:9000>

You should now see **TheHive login page** 🎉



If the page doesn't load, your firewall might be blocking port 9000.

Allow it using: `sudo ufw allow 9000`

To find the **latest default credentials** or learn how to perform the initial admin setup, refer to the official StrangeBee documentation:

👉 [Perform Initial Setup as Admin — TheHive Docs](#)

This ensures you always use the **most up-to-date credentials and security setup** recommended by TheHive team.

Integrate Windows 11 VM with Wazuh

Now that both **Wazuh** and **TheHive** servers are running on your DigitalOcean VMs, you can connect your **Windows 11 VM (in VMware)** as a Wazuh agent.

Access the Wazuh Dashboard

Inside your Windows 11 VM:

1. Open your browser and go to `https://{{your-wazuh-vm-ip}}`
2. Log in using your Wazuh dashboard credentials.

Add a New Agent

1. In the Wazuh dashboard, click “Deploy new agent.”

The screenshot shows the Wazuh dashboard with the 'Deploy new agent' button highlighted in red. The dashboard includes sections for AGENTS SUMMARY, LAST 24 HOURS ALERTS, ENDPOINT SECURITY, THREAT INTELLIGENCE, SECURITY OPERATIONS, and CLOUD SECURITY.

2. Select **Windows** as the agent type.
3. In **Server address**, enter your Wazuh server IP.
4. Assign a friendly **Agent name** such as **Windows-11**.

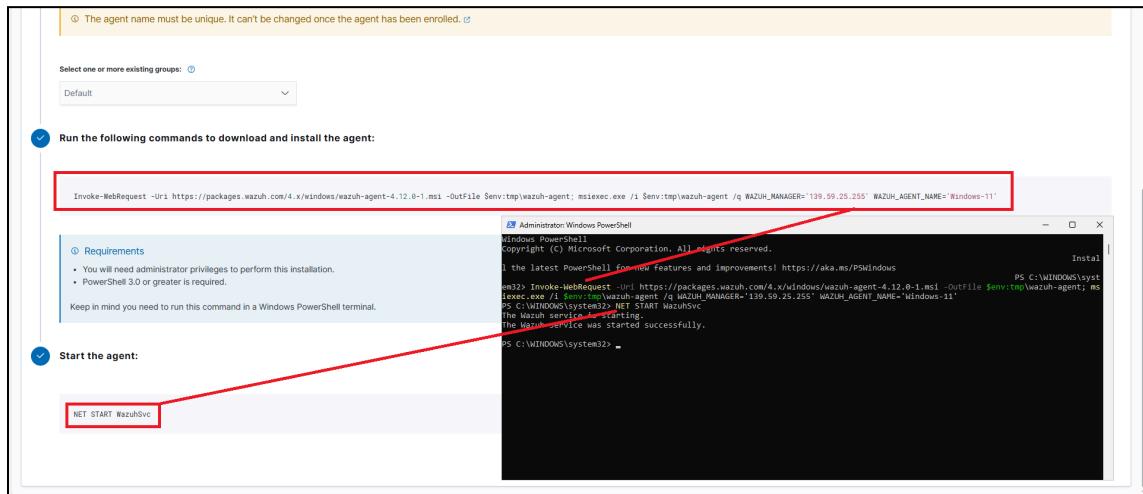
The screenshot shows the 'Deploy new agent' configuration page. The 'Select the package to download and install on your system' section has the 'WINDOWS' tab selected, highlighted with a red box. The 'Server address' field contains '139.59.25.255', and the 'Optional settings' field contains 'Windows-11'.

5. Copy the **PowerShell commands** shown on the screen.

Install the Agent on Windows

1. Open **PowerShell as Administrator** in your Windows VM.
2. Paste and run the copied commands.

3. Return to the Wazuh dashboard and refresh the Agents Summary page.



You should now see **Active (1)** under agents.

The screenshot shows the Wazuh dashboard. In the 'AGENTS SUMMARY' section, there is a green circle icon with a white outline. To its right, a red box highlights the 'Active (1)' status. Below this, four severity levels are listed: Critical severity (0), High severity (0), Medium severity (2,014), and Low severity (17,190). The dashboard also includes sections for ENDPOINT SECURITY (Configuration Assessment, Malware Detection, File Integrity Monitoring) and THREAT INTELLIGENCE (Threat Hunting, MITRE ATT&CK, Vulnerability Detection).

If the agent does not appear active:

On your Wazuh VM, allow the following ports:

```
sudo ufw allow 1514  
sudo ufw allow 1515
```

On your Windows VM, restart the Wazuh service: `Restart-Service -Name "wazuh"`
Refresh the dashboard again. The agent should now appear as active.

Configure Sysmon Telemetry in Wazuh

To enhance visibility, integrate **Sysmon** logs with your Windows Wazuh agent.

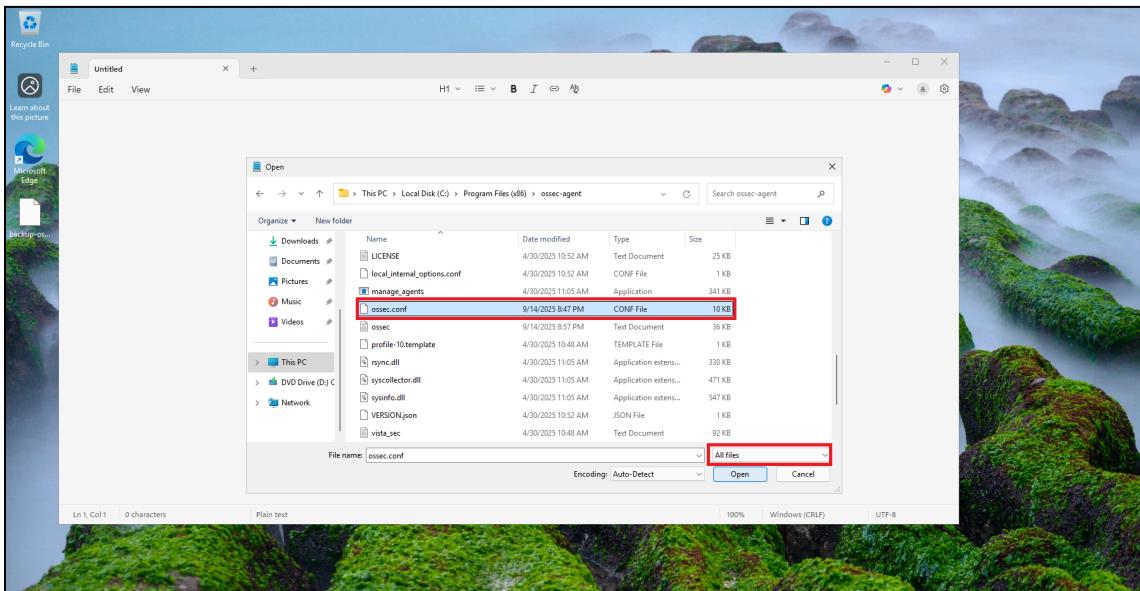
Backup Configuration

Make a backup of your current Wazuh agent config file: **C:\Program Files (x86)\ossec-agent\ossec.conf**

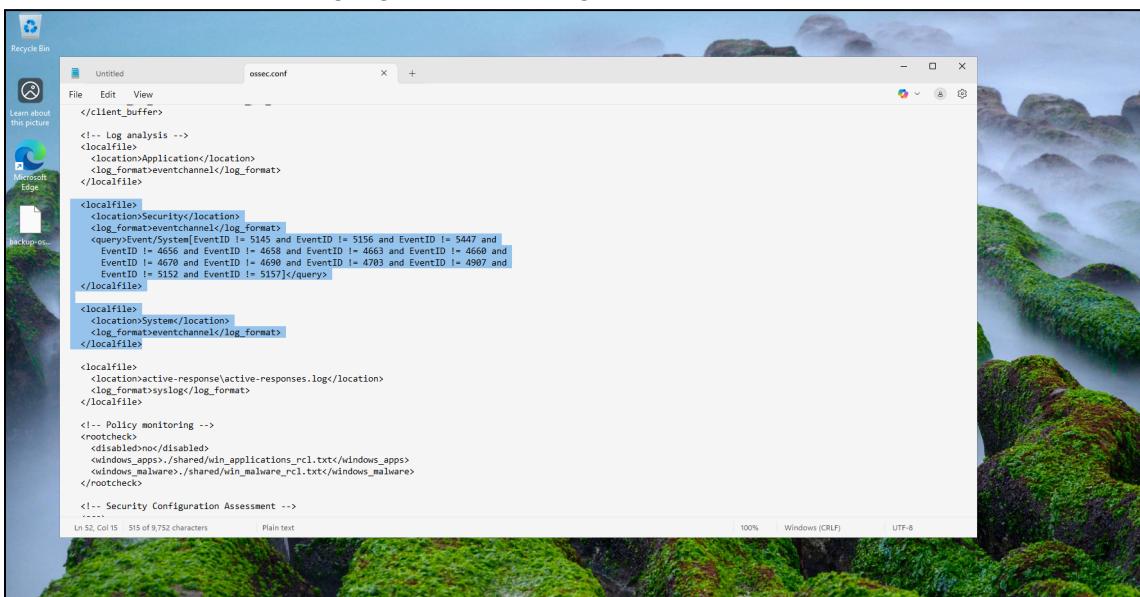
Copy it to another location (for example, your Desktop).

Edit Configuration

1. Open **Notepad as Administrator**.
2. Press **Ctrl + O** and navigate to: **C:\Program Files (x86)\ossec-agent**
3. Open **ossec.conf**.

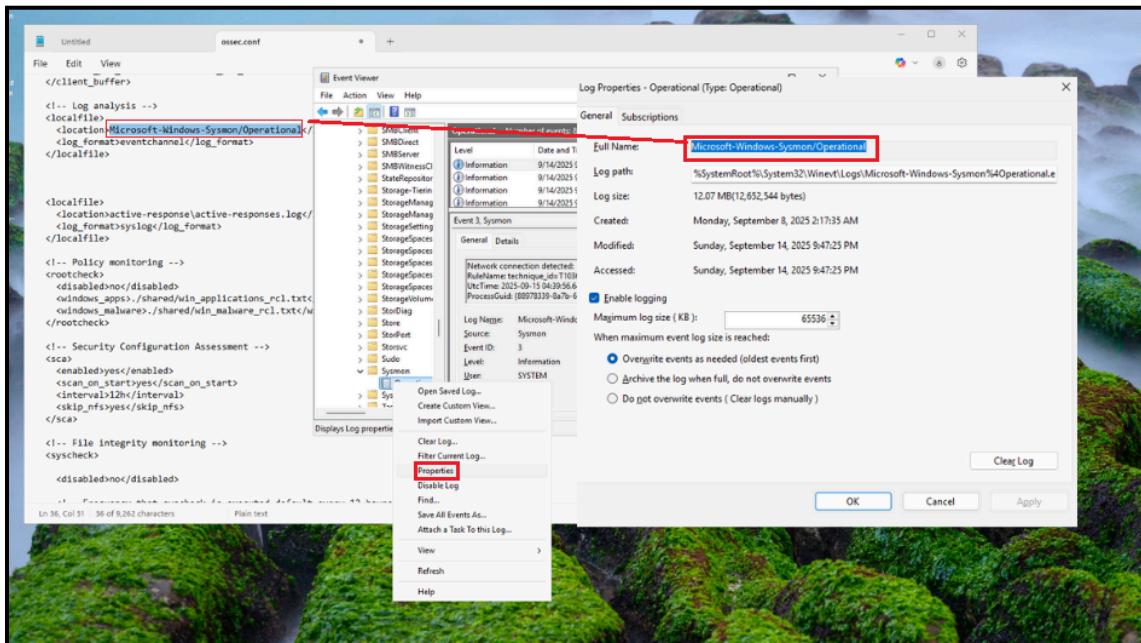


4. Locate and remove the highlighted default log entries.

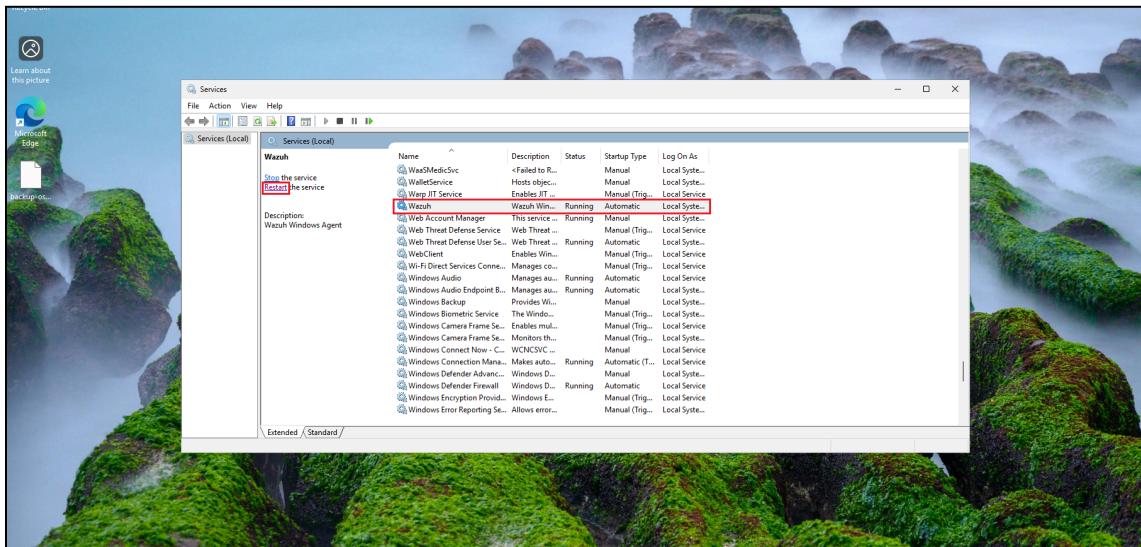


Add Sysmon Logs

1. Open Event Viewer → Applications and Services Logs → Microsoft → Windows → Sysmon → Operational.
2. Right-click Operational → Properties → copy the Full Name.
3. Paste this log path inside your `ossec.conf` file under the `<localfile>` section.

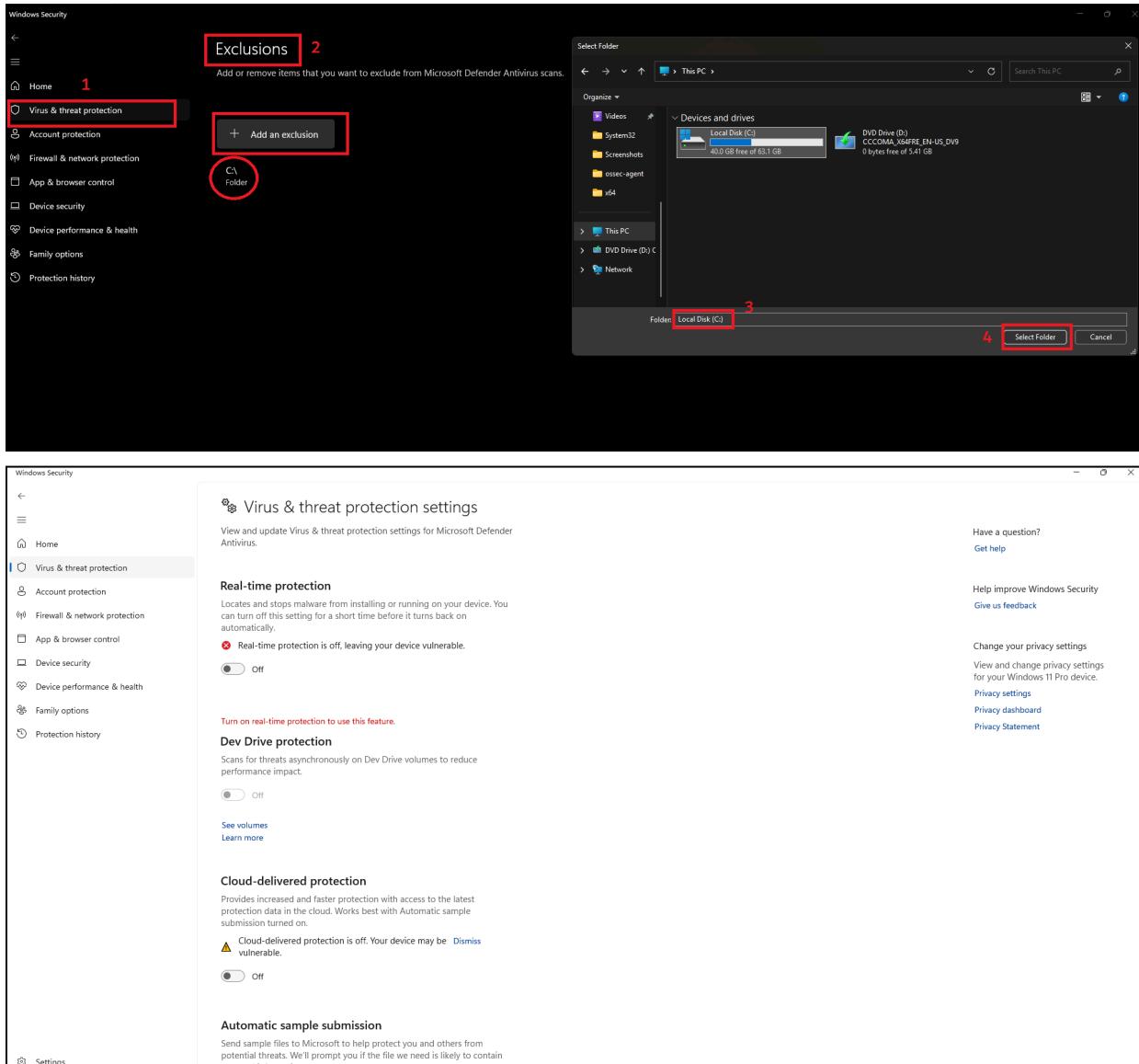


4. Save the file and restart the Wazuh service



Disable Windows Defender (for Lab Purposes Only)

1. Open Windows Security → Virus & threat protection.
2. Scroll down to Exclusions → Add or remove exclusions.
3. Click Add an exclusion → Folder → select C:\

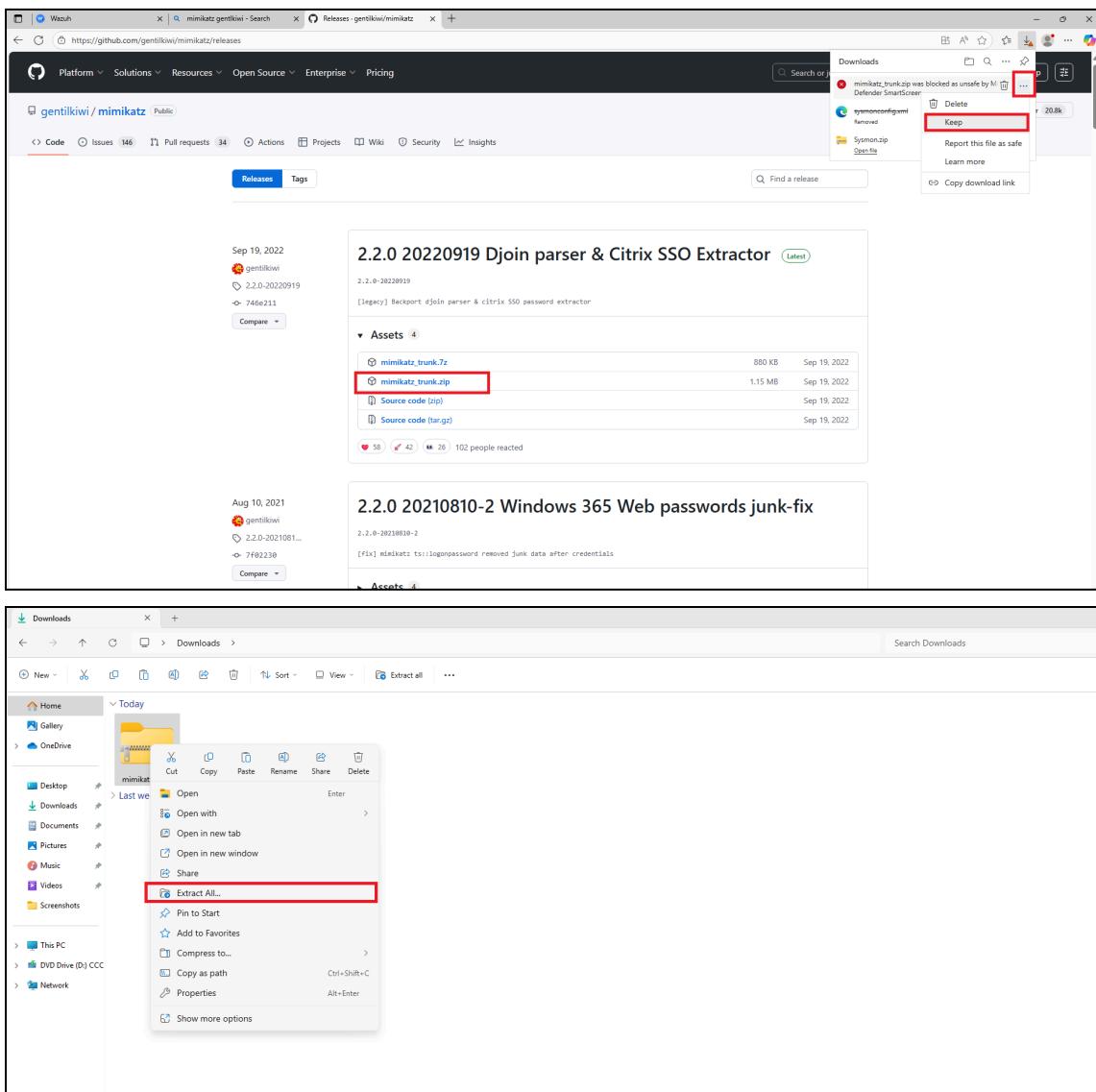


You have now successfully connected your **Windows 11 VM** to **Wazuh** and enabled **Sysmon telemetry**.

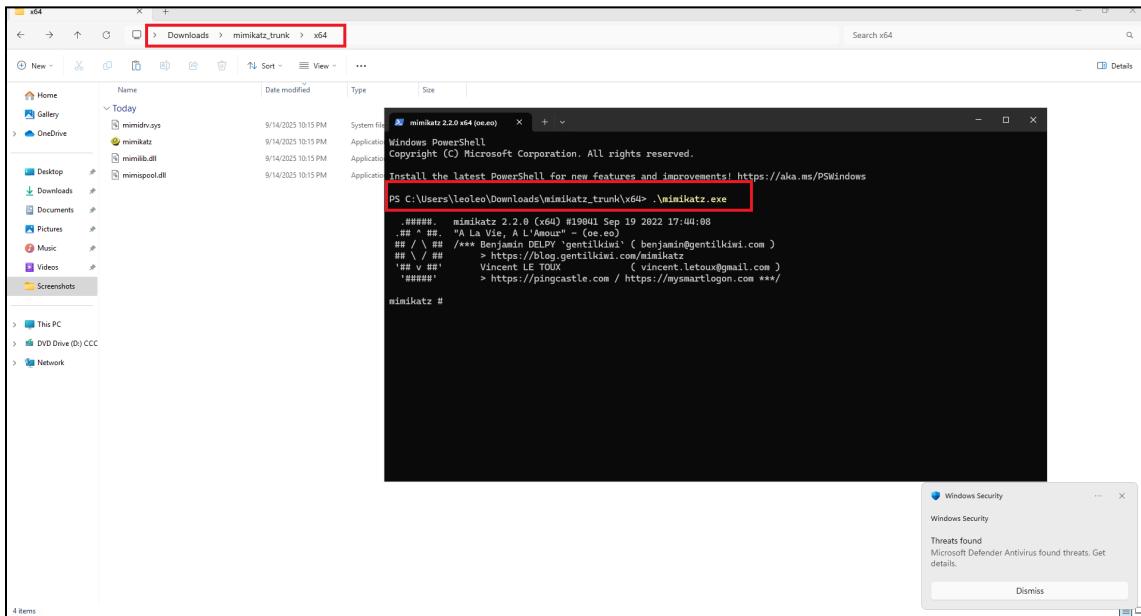
Install Mimikatz

1. Visit the official Mimikatz GitHub:
 - a. <https://github.com/gentilkiwi/mimikatz/releases>

2. Download and extract the latest release.



3. Open Command Prompt or PowerShell inside the x64 folder.



Now Mimikatz is ready for use in your test environment.

Configure Wazuh to Detect Mimikatz Activity

In your **Wazuh VM**, open and verify the logging configuration: `nano /var/ossec/etc/ossec.conf`

Ensure these lines are set to:

```
<logall>yes</logall>
<logall_json>yes</logall_json>
```

```
GNU nano 7.2                                     /var/ossec/etc/ossec.conf *
<!--
Wazuh - Manager - Default configuration for ubuntu 24.04
More info at: https://documentation.wazuh.com
Mailing list: https://groups.google.com/forum/#!forum/wazuh
-->

<ossec_config>
  <global>
    <jsonout_output>yes</jsonout_output>
    <alerts_log>yes</alerts_log>
    <logall>yes</logall>
    <logall_json>yes</logall_json>
    <email_notification>no</email_notification>
    <smtp_server>smtp.example.wazuh.com</smtp_server>
    <email_from>wazuh@example.wazuh.com</email_from>
    <email_to>recipient@example.wazuh.com</email_to>
    <email_maxperhour>12</email_maxperhour>
    <email_log_source>alerts.log</email_log_source>
    <agents_disconnection_time>10m</agents_disconnection_time>
    <agents_disconnection_alert_time>0</agents_disconnection_alert_time>
    <update_check>yes</update_check>
  </global>

  <alerts>
    <log_alert_level>3</log_alert_level>
    <email_alert_level>12</email_alert_level>
  </alerts>

  <!-- Choose between "plain", "json", or "plain,json" for the format of internal logs -->
  <logging>
    <log_format>plain</log_format>

^G Help          ^O Write Out      ^W Where Is      ^K Cut           ^T Execute       ^C Location      M-U Undo        M-A Set Mark
^X Exit          ^R Read File       ^\ Replace       ^U Paste         ^J Justify       ^/ Go To Line    M-E Redo        M-G Copy
```

Save the file and restart Wazuh: `systemctl restart wazuh-manager.service`

Check `/var/ossec/logs/archives/` logs should now be generated there.

Adjust Filebeat Configuration: `nano /etc/filebeat/filebeat.yml`

```

GNU nano 7.2
# Wazuh - Filebeat configuration file
/etc/filebeat/filebeat.yml *

output.elasticsearch.hosts:
- 127.0.0.1:9200
#     - <elasticsearch_ip_node_2>:9200
#     - <elasticsearch_ip_node_3>:9200

output.elasticsearch:
  protocol: https
  username: ${username}
  password: ${password}
  ssl.certificateAuthorities:
    - /etc/filebeat/certs/root-ca.pem
  ssl.certificate: "/etc/filebeat/certs/wazuh-server.pem"
  ssl.key: "/etc/filebeat/certs/wazuh-server-key.pem"
setup.template.json.enabled: true
setup.template.json.path: '/etc/filebeat/wazuh-template.json'
setup.template.json.name: 'wazuh'
setup.ilm.overwrite: true
setup.ilm.enabled: false

filebeat.modules:
- module: wazuh
  alerts:
    enabled: true
  archives:
    enabled: true

logging.level: info
logging.to_files: true
logging.files:
  path: /var/log/filebeat

^G Help      ^O Write Out   ^W Where Is   ^K Cut          ^T Execute   ^C Location   M-U Undo   M-A Set Mark
^X Exit      ^R Read File   ^A Replace   ^U Paste        ^J Justify   ^I Go To Line M-E Redo   M-O Copy

```

save (**Ctrl + X** → **Y** → **Enter**), and restart: `systemctl restart filebeat.service`

Now return to your **Wazuh dashboard** and create a **new index**.

The screenshot shows the 'Index patterns' section of the 'Dashboards Management' interface. On the left, there's a sidebar with 'Dashboards Management' at the top, followed by 'Index patterns' (which is highlighted with a red box), 'Data sources', 'Saved objects', and 'Advanced settings'. The main area has a title 'Index patterns' and a sub-instruction 'Create and manage the index patterns that help you retrieve your data from OpenSearch.' Below this is a search bar with 'Search...'. A table lists index patterns: 'wazuh-alerts-*' (Default) and 'wazuh-monitoring-*'. To the right of the table is a 'Data Source Connection' section with three empty lines. At the bottom right of the main area is a blue button with white text '+ Create index pattern' (also highlighted with a red box). Navigation controls at the bottom right include arrows for page navigation and a 'Rows per page: 10' dropdown.

The screenshot shows the 'Create index pattern' page. The sidebar on the left is identical to the one in the first screenshot. The main area has a title 'Create index pattern' and a sub-instruction 'An index pattern can match a single source, for example, filebeat-4-3-22, or multiple data sources, filebeat-*.' Below this is a link 'Read documentation'. The next section is 'Step 1 of 2: Define an index pattern'. It has an 'Index pattern name' input field containing 'wazuh-archives-*' (highlighted with a red box) and a 'Next step >' button (also highlighted with a red box). There's also a note: 'Use an asterisk (*) to match multiple indices. Spaces and the characters \, /, ?, ", <, >, | are not allowed.' Below the input field is a checkbox 'Include system and hidden indices' and a status message 'Your index pattern matches 1 source.' (preceded by a green checkmark). At the bottom, it shows 'wazuh-archives-4.x-2025.09.15' and an 'Index' button. Navigation controls at the bottom right are identical to the first screenshot.

Create index pattern

An index pattern can match a single source, for example, `filebeat-4-3-22`, or **multiple** data sources, `filebeat-*`.

[Read documentation](#)

Step 2 of 2: Configure settings

Specify settings for your `wazuh-archives-*` index pattern.

Select a primary time field for use with the global time filter.

Time field Refresh

[Show advanced settings](#)

[Back](#) Create index pattern

Discover

`wazuh-archives-*` |

Index patterns

- wazuh-alerts-*
- wazuh-archives-***
- wazuh-monitoring-*
- wazuh-statistics-*

_index

@timestamp

agent.id

agent.ip

agent.name

data.dstuser

data.euid

data.srchip

data.srcport

data.srcuser

data.tty

data.uid

data.win.eventdata.

company

data.win.eventdata.

eventual_hits

mimikatz DQL Last 24 hours Show dates Refresh

Add filter

2 hits

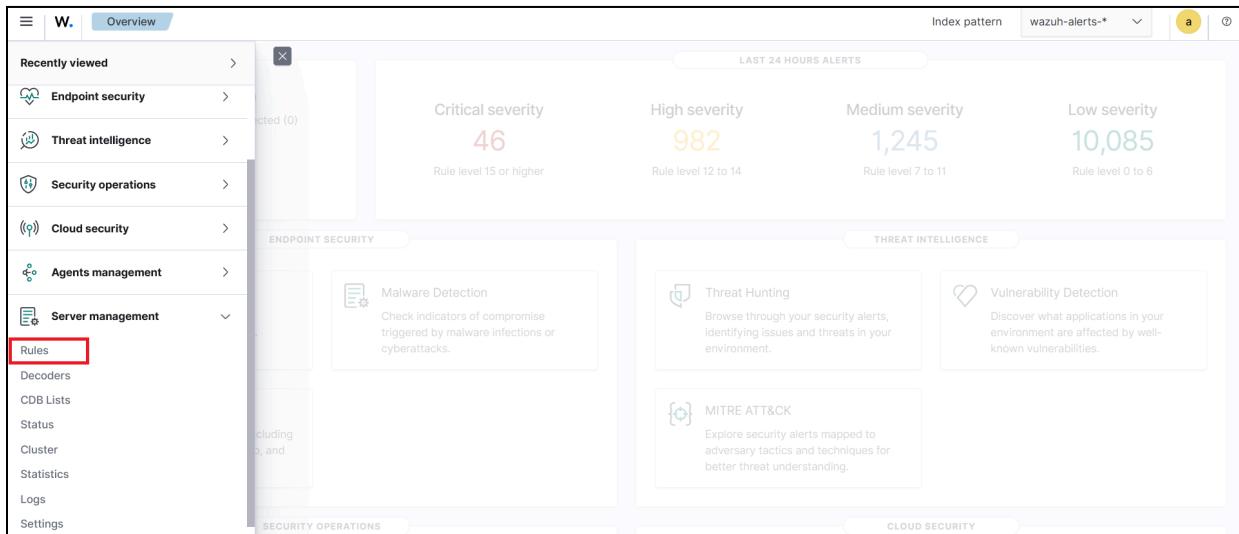
Sep 14, 2025 @ 11:10:16.336 - Sep 15, 2025 @ 11:10:16.336 per Auto

Count

Time _source

Time	_source
Sep 15, 2025 @ 11:09:45	agent.ip: 192.168.184.129 agent.name: Windows-11 agent.id: 001 manager.name: Wazuh data.win.eventdata.originalFileName: mimikatz.exe data.win.eventdata.image: C:\Users\leoleo\Downloads\mimikatz_trunk\x64\mimikatz.exe data.win.eventdata.product: mimikatz data.win.eventdata.imageLoaded: C:\Users\leoleo\Downloads\mimikatz_trunk\x64\mimikatz.exe data.win.eventdata.description: mimikatz for Windows data.win.eventdata.signed: false
Sep 15, 2025 @ 11:09:50.738	agent.ip: 192.168.184.129 agent.name: Windows-11 agent.id: 001 manager.name: Wazuh data.win.eventdata.originalFileName: mimikatz.exe data.win.eventdata.image: C:\Users\leoleo\Downloads\mimikatz_trunk\x64\mimikatz.exe data.win.eventdata.product: mimikatz data.win.eventdata.parentProcessGuid: {88978339-ad98-68c7-6006-000000000000} data.win.eventdata.description: mimikatz for Windows data.win.eventdata.logonGuid: {88978339-88a}

Create a Detection Rule for Mimikatz



The screenshot shows the Wazuh UI Rules page. It displays a table titled 'Rules (4,490)' with columns: ID, Description, Groups, Regulatory compliance, Level, File, and Path. The 'File' column contains links to XML files like '0010-rules_config.xml'. A red box highlights the 'Custom rules' button in the top right corner of the table header.

ID	Description	Groups	Regulatory compliance	Level	File	Path
1	Generic template for all syslog rules.	syslog		0	0010-rules_config.xml	ruleset/rules
2	Generic template for all firewall rules.	firewall		0	0010-rules_config.xml	ruleset/rules
3	Generic template for all ids rules.	ids		0	0010-rules_config.xml	ruleset/rules
4	Generic template for all web rules.	web-log		0	0010-rules_config.xml	ruleset/rules
5	Generic template for all web proxy rules.	squid		0	0010-rules_config.xml	ruleset/rules
6	Generic template for all windows rules.	windows		0	0010-rules_config.xml	ruleset/rules
7	Generic template for all wazuh rules.	ossec		0	0010-rules_config.xml	ruleset/rules
200	Grouping of wazuh rules.	wazuh		0	0016-wazuh_rules.xml	ruleset/rules

Add the following custom rule to your ruleset:

```
<group name="local,syslog,sshd,">

<rule id="100001" level="5">
  <if_sid>5716</if_sid>
  <srcip>1.1.1.1</srcip>
  <description>sshd: authentication failed from IP 1.1.1.1.</description>
  <group>authentication_failed, pci_dss_10.2.4, pci_dss_10.2.5</group>
</rule>

<rule id="100002" level="15">
  <if_group>sysmon_event1</if_group>
  <field name="win.eventdata.originalFileName">
```

```

type="pcre2">(?i)mimikatz\.exe</field>
    <description>Suspicious Mimikatz activity detected on
system</description>
    <mitre>
        <id>T1003</id>
    </mitre>
</rule>

</group>

```

W. Rules

Index pattern: wazuh-alerts-*

My_rule.xml

Save

```

<group name="local,syslog, sshd,">
    <rule id="100001" level="5">
        <if_sid>5716</if_sid>
        <srcip>1.1.1.1</srcip>
        <description>sshd: authentication failed from IP 1.1.1.1.</description>
        <group>authentication_failed, pci_dss_10.2.4, pci_dss_10.2.5, </group>
    </rule>
    <rule id="100002" level="15">
        <if_group>sysmon_event1</if_group>
        <field name="win.eventdata.originalFileName" type="pcre2">(?i)mimikatz\.exe</field>
        <description>Suspicious Mimikatz activity detected on the system</description>
        <mitre>
            <id>T1003</id>
        </mitre>
    </rule>
</group>

```

W. Rules

Index pattern: wazuh-alerts-*

My_rule.xml

Save

Changes will not take effect until a restart is performed.

Restart

```

<group name="local,syslog, sshd,">
    <rule id="100001" level="5">
        <if_sid>5716</if_sid>
        <srcip>1.1.1.1</srcip>
        <description>sshd: authentication failed from IP 1.1.1.1.</description>
        <group>authentication_failed, pci_dss_10.2.4, pci_dss_10.2.5, </group>
    </rule>
    <rule id="100002" level="15">
        <if_group>sysmon_event1</if_group>
        <field name="win.eventdata.originalFileName" type="pcre2">(?i)mimikatz\.exe</field>
        <description>Suspicious Mimikatz activity detected on the system</description>
        <mitre>
            <id>T1003</id>
        </mitre>
    </rule>
</group>

```

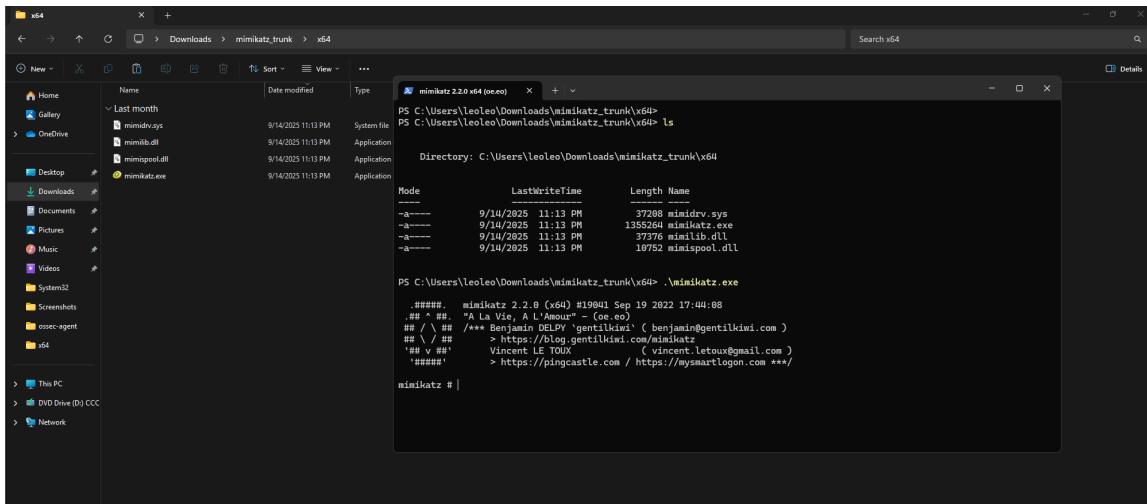
Rule Explanation (In Simple Terms)

- **Rule 1:** Detects repeated SSH authentication failures from IP 1.1.1.1.

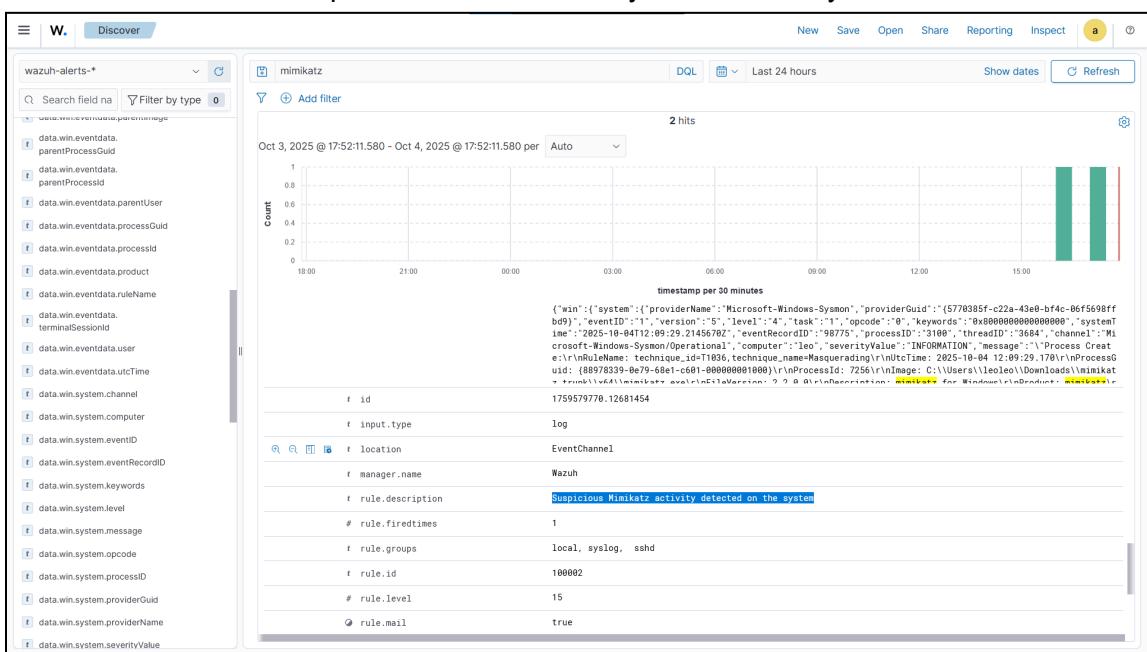
- **Rule 2:** Detects Sysmon Event ID 1 (process creation) if the executable name is **mimikatz.exe**.
→ This triggers a **level 15 alert** with MITRE technique **T1003 (Credential Dumping)**.

Validate the Rule

1. Run Mimikatz again in your Windows VM.



2. Go to your **Explore** → **Discover** → select **wazuh-alerts** → search **mimikatz**. You should see an alert labeled “Suspicious Mimikatz activity detected on system”



Automating SOC Alerts with Shuffle

Now that detection is working, let's automate the response workflow using **Shuffle**.

Create a Shuffle Account

Go to <https://shuffler.io> → **Sign up / Log in** → In the sidebar, click **Workflows** → **+ Create Workflow** → name it **soc-workflow**.

Add Webhook Integration

- From the **left panel**, drag **Webhook** onto the canvas.
- Copy its **Webhook URI**.

Now, go back to your **Wazuh VM** and open the config file: `sudo nano /var/ossec/etc/ossec.conf`

Add this under the `</global>` tag:

```
<integration>
  <name>shuffle</name>
  <hook_url>{your-webhook-URI}</hook_url>
  <rule_id>100002</rule_id>
  <alert_format>json</alert_format>
</integration>
```

```

GNU nano 7.2
/var/ossec/etc/ossec.conf *
Wazuh - Manager - Default configuration for ubuntu 24.04
More info at: https://documentation.wazuh.com
Mailing list: https://groups.google.com/forum/#!forum/wazuh
-->

<ossec_config>
  <global>
    <jsonout_output>yes</jsonout_output>
    <alerts_log>yes</alerts_log>
    <logall>yes</logall>
    <logall_json>yes</logall_json>
    <email_notification>no</email_notification>
    <smtp_server>smtp.example.wazuh.com</smtp_server>
    <email_from>wazuh@example.wazuh.com</email_from>
    <email_to>recipient@example.wazuh.com</email_to>
    <email_maxperhour>12</email_maxperhour>
    <email_log_source>alerts.log</email_log_source>
    <agents_disconnection_time>10m</agents_disconnection_time>
    <agents_disconnection_alert_time>0</agents_disconnection_alert_time>
    <update_check>yes</update_check>
  </global>

  <integration>
    <name>shuffle</name>
    <hook_url>https://shuffler.io/api/v1/hooks/webhook\_c569ced1-80f5-4d2d-875e-2ecf85e14abb</hook_url>
    <rule_id>100002</rule_id>
    <alert_format>json</alert_format>
  </integration>

  <alerts>
    <log_alert_level>3</log_alert_level>

```

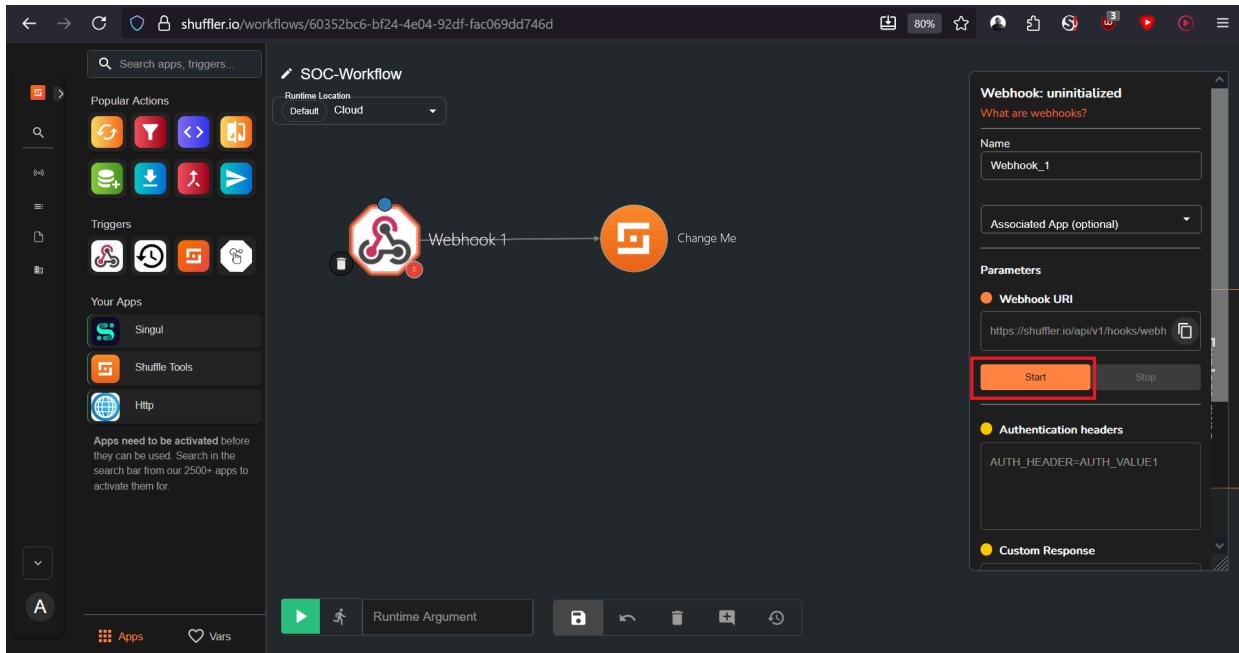
File Name to Write: /var/ossec/etc/ossec.conf

File Name to Write: /var/ossec/etc/ossec.conf

M-G Help M-D DOS Format M-A Append M-B Backup File
 ^C Cancel M-M Mac Format M-P Prepend M-T Browse

Save and restart the service: `sudo systemctl restart wazuh-manager.service`

In Shuffle, click **Start Webhook** to begin listening.



Test the Webhook

Run **Mimikatz** again in your Windows VM.

The screenshot shows a Windows File Explorer window with the path 'Downloads > mimikatz_trunk > x64'. Inside this folder, there are four files: minidrv.sys, minilib.dll, minispool.dll, and mimikatz.exe. The mimikatz.exe file was recently modified on 9/14/2025 at 11:13 PM. A PowerShell window titled 'mimikatz 2.2.0 x64 (oe.eo)' is running in the background, showing the command 'ls' and its output:

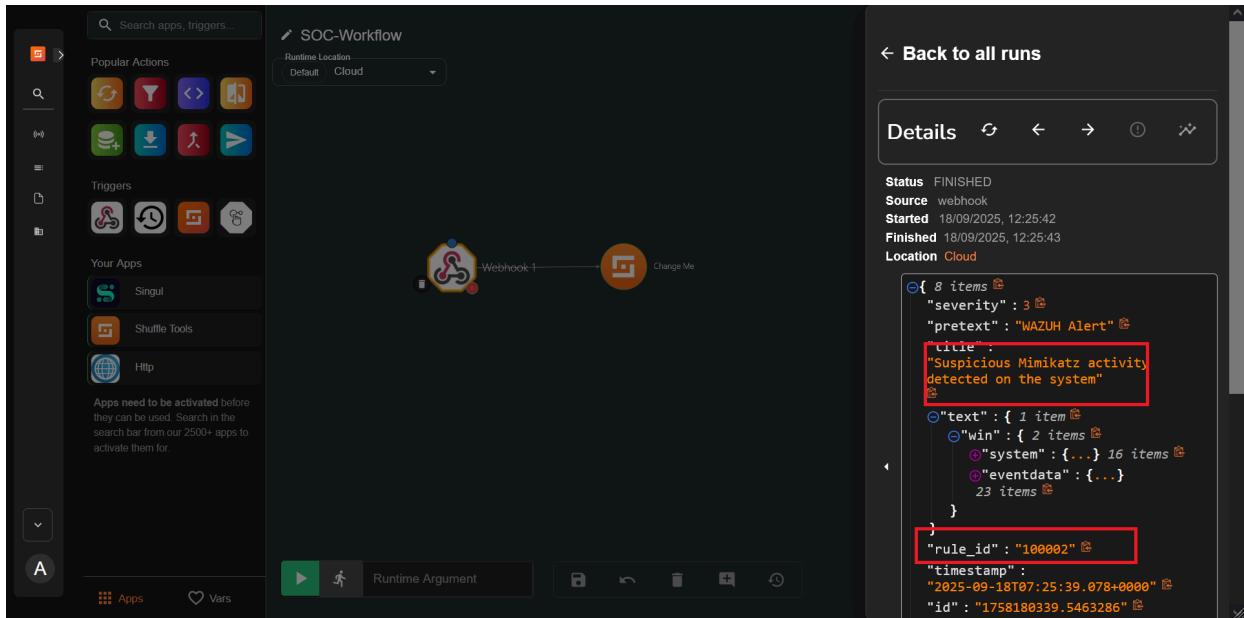
```
PS C:\Users\leoleo\Downloads\mimikatz_trunk\x64> ls
PS C:\Users\leoleo\Downloads\mimikatz_trunk\x64> Directory: C:\Users\leoleo\Downloads\mimikatz_trunk\x64

Mode LastWriteTime Length Name
-- -- -- -- --
-a--- 9/14/2025 11:13 PM 37288 minidrv.sys
-a--- 9/14/2025 11:13 PM 1355264 mimikatz.exe
-a--- 9/14/2025 11:13 PM 37376 minilib.dll
-a--- 9/14/2025 11:13 PM 10752 minispool.dll

PS C:\Users\leoleo\Downloads\mimikatz_trunk\x64> .\mimikatz.exe
.####. mimikatz 2.2.0 (x64) #19841 Sep 19 2022 17:44:08
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /* Benjamins DELPY 'gentilkiwi' ( benjamin@gentilkiwi.com )
## ' / \ ## > https://blog.gentilkiwi.com/mimikatz
'## v ##. Vincent LE TOUX
'## v ##. ( vincent.letoux@gmail.com )
'## v ##. > https://pingcastle.com / https://mysmartlogon.com ***/
'#####'
```

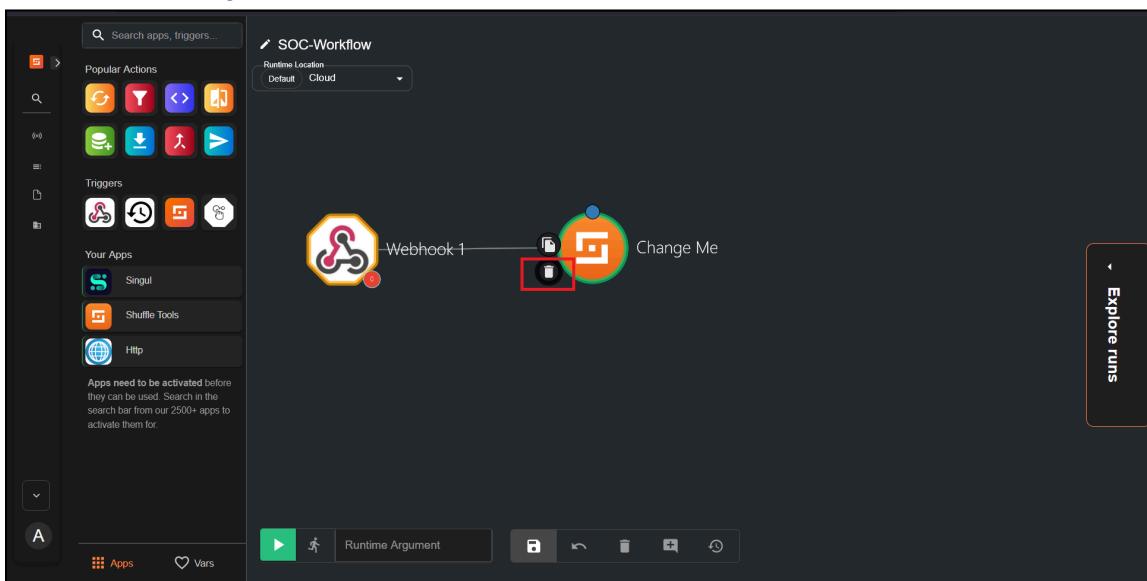
Then go to **Explore Runs** in Shuffle → you should see data received successfully.

The screenshot shows the Shuffle interface with the 'SOC-Workflow' template selected. On the left, there's a sidebar with sections for 'Popular Actions', 'Triggers', and 'Your Apps'. The main workspace shows a flowchart with a 'Webhook 1' node connected to a 'Charge Me' node. In the bottom right corner, there is a button labeled 'Explore runs' with a red rectangular box drawn around it.



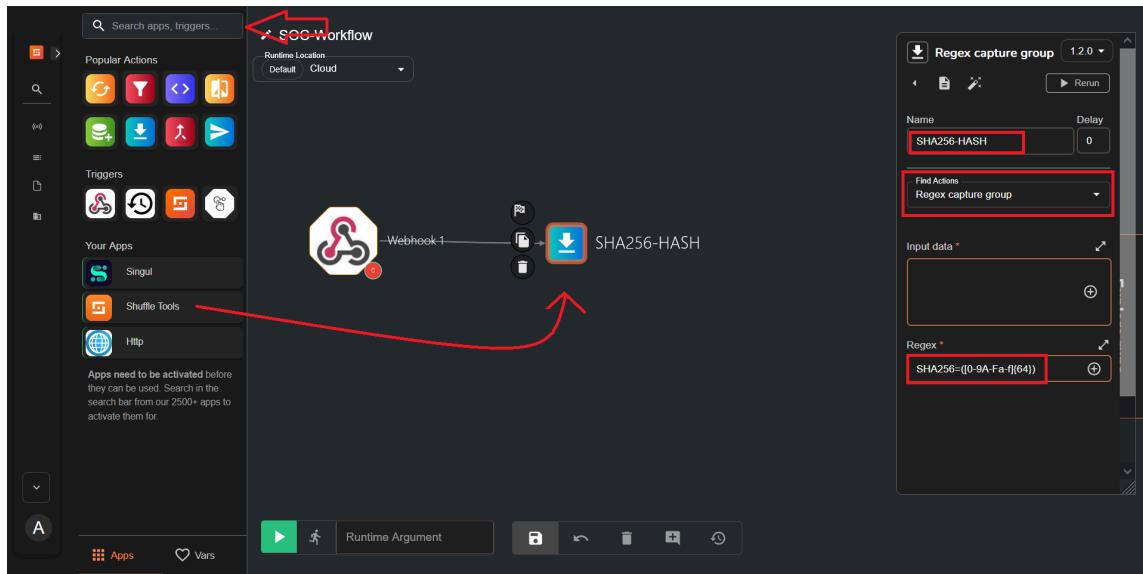
Add SHA256 Extraction

- Delete the “Change Me” icon.

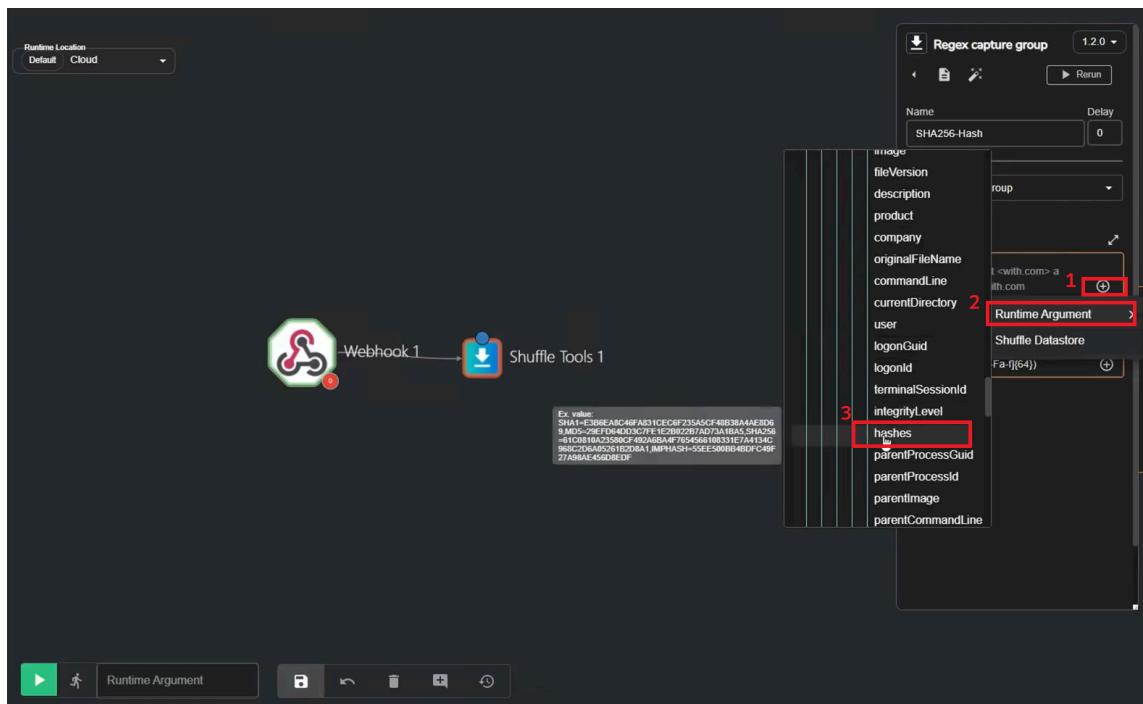


- Drag **Shuffle Tools** → rename it → under **Find Actions**, choose **Regex Capture Group**.
- Use this Regex:

SHA256=([0-9A-Fa-f]{64})



- For **Input data**, click the \oplus icon \rightarrow select **Runtime Argument** \rightarrow **hashes**.



Add VirusTotal Integration

- Go to <https://www.virustotal.com> → Sign in → click your profile → **API Key** → copy it.

This screenshot shows the 'API KEY' section of the VirusTotal website. It displays a personal API key and a note about its confidentiality. Below the key is a 'Copy key to clipboard' button, which is highlighted with a red box. The page also includes 'API QUOTA ALLOWANCES FOR YOUR USER' information and links to various API client libraries.

- In Shuffle, drag the **VirusTotal** app → click **Authenticate** → paste your API key.

This screenshot shows the Shuffle platform interface. On the left, the 'Your Apps' section lists the 'VirusTotal v3' app, which is highlighted with a red arrow pointing to it from the left. The main workspace shows a workflow starting with a 'Webhook 1' node, followed by a 'SHA256-HASH' download node, and finally a 'VirusTotal-Auth' node. To the right, a detailed configuration panel for the 'Get a hash report' action is shown, with the 'Authenticate Virustotal v3' step highlighted with a red box. The configuration fields include 'Apikey', 'Url' (set to https://www.virustotal.com), and 'Id' (set to Value).

SOC-Workflow

Authentication for Virustotal v3

What is app authentication?
These are required fields for authenticating with Virustotal_v3

Label for you to remember: Virustotal_v3-Auth

apikey: (Red box)

url: https://www.virustotal.com

Submit

Virustotal V3

Authentication

For authenticating with the API you must include the x-apikey header with your personal API key in all your requests. Your API key can be found in your VirusTotal account user menu:



Profile

SOC-Workflow

Runtime Location: Default / Cloud

Webhook 1 → SHA256-HASH → Get a hash report

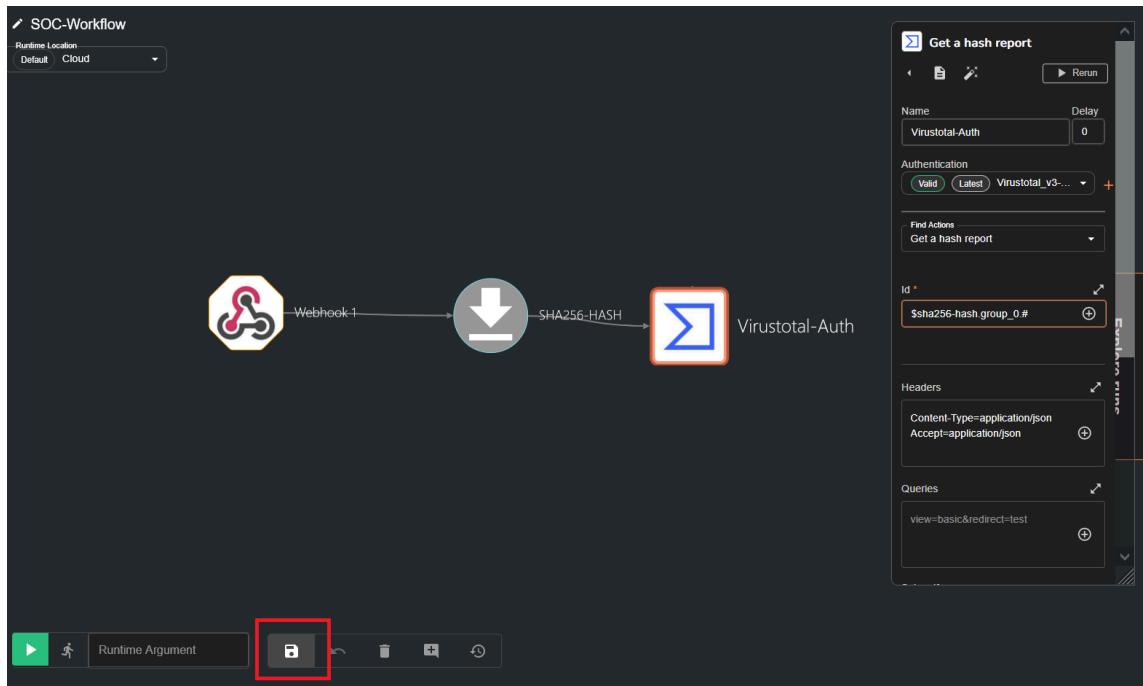
SHA256-HASH: success, group_0, list, found (Red box)

Get a hash report: Name: Virustotal-Auth, Delay: 0, Authentication: Virustotal_v3-Auth (Red box)

Runtime Argument: SHA256-HASH (Red box)

Runtime Argument: application/json

- Connect it with the SHA256 output.



Save your workflow and refresh the page.

Test again by running Mimikatz in your Windows VM → you should see "success": true and "status": 200 in Shuffle.

The screenshot shows a Windows File Explorer window on the left and a PowerShell terminal window on the right.

File Explorer:

- Path: Downloads > mimikatz_trunk > x64
- Files listed:
 - minidrv.sys
 - minilib.dll
 - mimispool.dll
 - mimikatz.exe

PowerShell Terminal:

```

PS C:\Users\leoleo\Downloads\mimikatz_trunk\x64> ls
Directory: C:\Users\leoleo\Downloads\mimikatz_trunk\x64

Mode                LastWriteTime         Length Name
--                -<->                 -<->     -
-a----   9/14/2025 11:13 PM          37288 minidrv.sys
-a----   9/14/2025 11:13 PM        1355264 mimikatz.exe
-a----   9/14/2025 11:13 PM          37376 minilib.dll
-a----   9/14/2025 11:13 PM          10752 mimispool.dll

PS C:\Users\leoleo\Downloads\mimikatz_trunk\x64> .\mimikatz.exe
#####
# mimikatz 2.2.0 (x64) #19841 Sep 19 2022 17:44:08
## ^ ##
# "La Vie, A L'Amour" - (oe.eo)
## / ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > https://blog.gentilkiwi.com/mimikatz
## v ## Vincent LE TOUX
## v ## ( vincent.letoux@gmail.com )
## v ## > https://pingcastle.com / https://mysmartlogon.com ***/
#####
mimikatz #

```

The screenshot shows the Zapier interface with a workflow titled "SOC-Workflow". The workflow starts with a "Webhook 1" trigger, followed by a "SHA256-HASH" action, and ends with a "Virustotal-Auth" action. The "SHA256-HASH" action's details pane shows a JSON response with a "success" key highlighted in red. The "Virustotal-Auth" action's details pane shows a JSON response with a "status" key highlighted in red.

Add TheHive Integration

Go to your TheHive dashboard: <http://{your-thehive-vm-ip}:9000>

1. Create a new Organization → click +.

The screenshot shows the TheHive interface with an "Organisation List" page. A modal window titled "Adding an Organisation" is open. The "Name" field is filled with "SOC-Workflow" and the "Confirm" button is highlighted with a red box.

2. Inside it, create a **User Account** → click + → fill in details → set password → Confirm.

The screenshot shows the 'Adding a User' dialog box over a background of the 'Organisation List / #SOC-Workflow / Users' page. The dialog has a red box around the 'Type' field set to 'Normal'. Other fields include 'Login' (leo@test.com), 'Name' (leo), 'Profile' (analyst), and 'Permissions' (a large list of API permissions). The 'Confirm' button at the bottom right is also highlighted with a red box.

3. Inside it, also create a **Service Account** → click + → fill in details → set password → Confirm.

The screenshot shows the 'Adding a User' dialog box over a background of the 'Organisation List / #SOC-Workflow / Users' page. The dialog has a red box around the 'Type' field set to 'Service'. Other fields include 'Login' (shuffle@test.com), 'Name' (lee), 'Profile' (analyst), and 'Permissions' (a large list of API permissions). The 'Confirm' button at the bottom right is also highlighted with a red box.

4. Go to **Service Accounts** → create → copy its API key

This instance uses a **Platinum License** for **Trial** purpose, and

lee **Active**

id -24704 **Created by** Default admin user
Created at 18/09/2025 13:23 **Updated at** 18/09/2025 13:23

Email

Type Service

Locked

MFA No

API Key

Create **Reveal** **Revoke**

Profile analyst
License required

Permissions

Delete user

This instance uses a **Platinum License** for **Trial** purpose, and

lee **Active**

id -24704 **Created by** Default admin user
Created at 18/09/2025 13:23 **Updated at** 18/09/2025 13:24

Email

Type Service

Locked

MFA No

API Key **JWbCka04H+sjz2/EEclWTezNizyG9N+j**

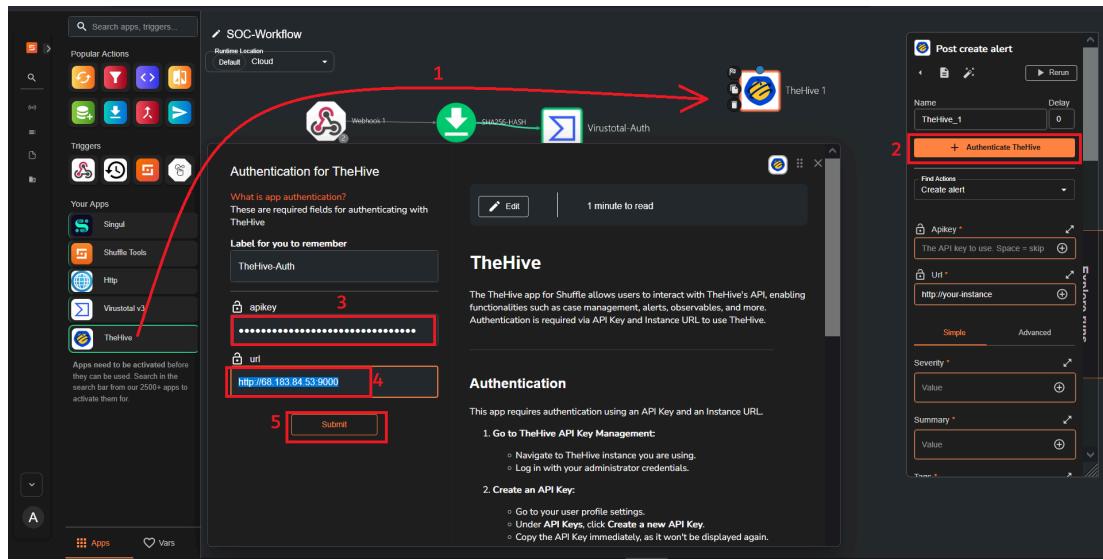
Renew **Reveal** **Revoke**

Profile analyst
License required

Delete user

Now in Shuffle:

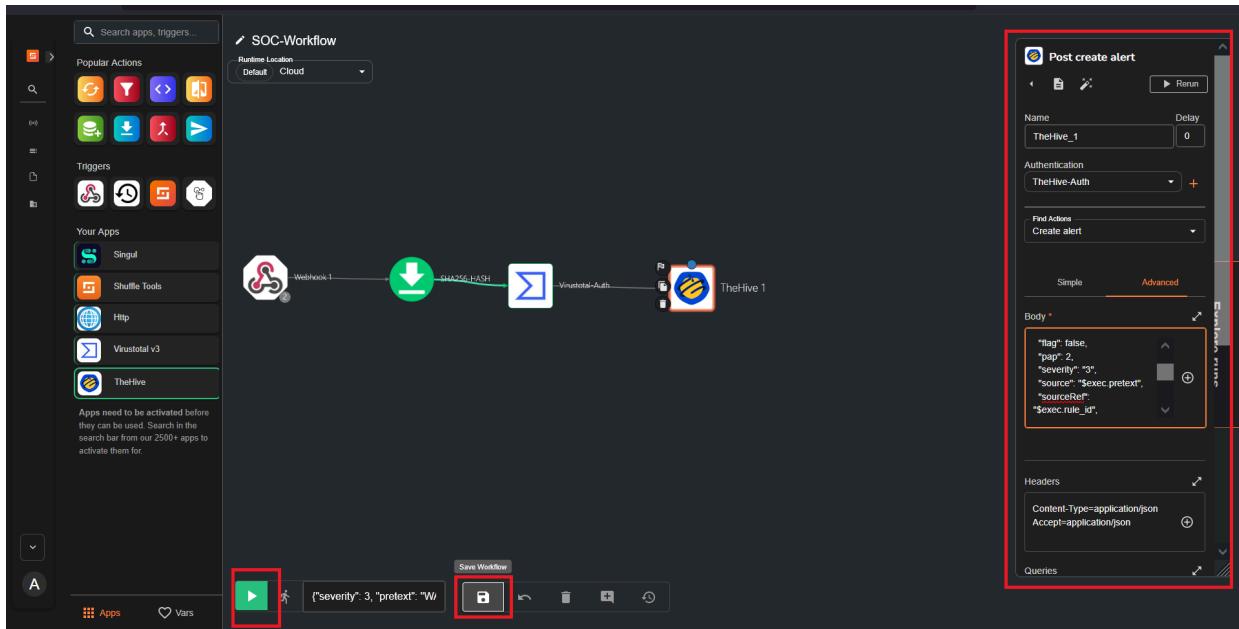
- Drag **TheHive** app → click **Authenticate** → use the service account API key.



- Connect it after VirusTotal.

Paste this body:

```
{
  "description": "$exec.title",
  "externallink": "${externallink}",
  "flag": false,
  "pap": 2,
  "severity": "3",
  "source": "$exec.pretext",
  "sourceRef": "$exec.rule_id",
  "status": "New",
  "summary": "Mimikatz usage detected on host: $exec.text.win.system.computer",
  "tags": ["T1003"],
  "title": "$exec.title",
  "tlp": "2",
  "type": "Internal"
}
```



Save and run the workflow → verify on TheHive that a **new alert** appears.

The screenshot shows two views of the Wazuh interface. The top view is a list of alerts, with one specific alert selected. The bottom view is a detailed view of that selected alert.

Alerts List View (Top):

- Header: This instance uses a **Platinum** License for Trial purpose, and will expire in **13** days. [Register now.](#)
- Toolbar: Alerts, Enter a case number, + Create Case, UK flag, Help, Logout, Refresh.
- Filter: default, Quick Filters, Export list.
- Table Headers: Status, Severity, Title, # Case, Type, Source, Reference, Details, Assignee, Dates, O., C., U.
- Alert Details (Selected):
 - Status: New (New a few seconds ago)
 - Type: Internal
 - Source: WAZUH Alert
 - Reference: T1003
 - Details: Observables 0, TTPs 0, Occurred date: 18/09/2025 13:43
 - Comments: None

Alert Detail View (Bottom):

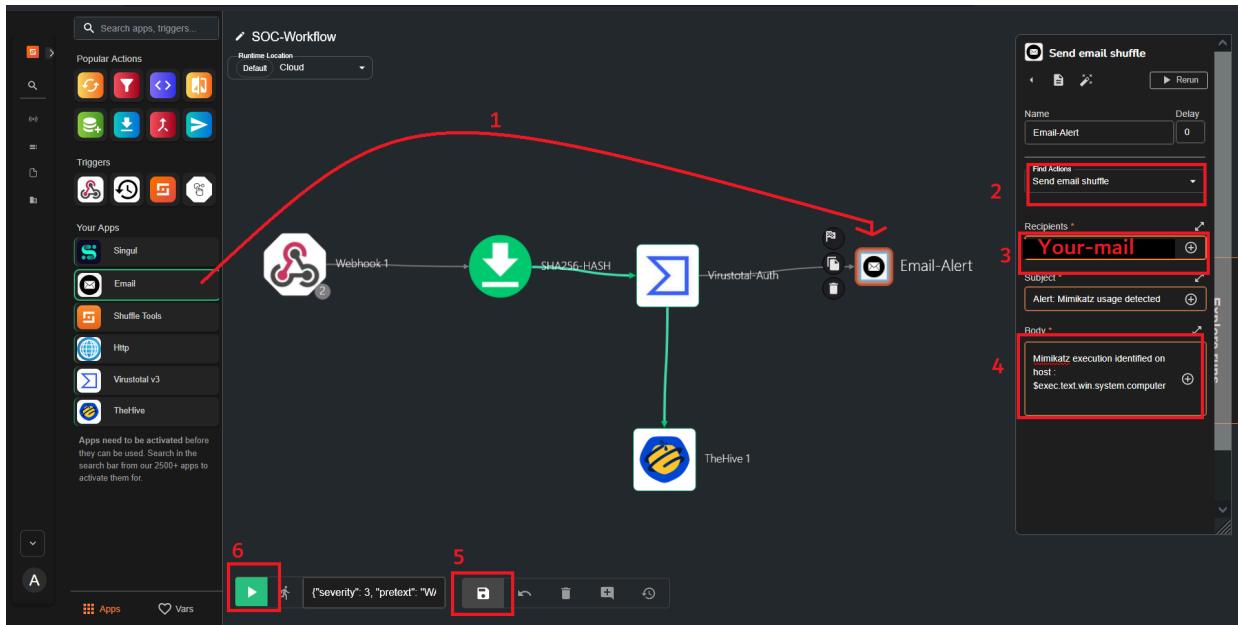
- Header: This instance uses a **Platinum** License for Trial purpose, and will expire in **13** days. [Register now.](#)
- Toolbar: Alerts / Internal (#1000...) / Description, Enter a case number, + Create Case, UK flag, Help, Logout, Refresh.
- Left Sidebar:
 - S: Suspicious Mimikatz activity detected on the system
 - id: ~4248
 - Created by: shuffle@test.com
 - Created at: 18/09/2025 13:43
 - Severity: HIGH
 - TLP: AMBER
 - PAP: AMBER
 - Assignee: Assign to me (Unassigned)
 - Source: WAZUH Alert
 - Reference: 100002
 - Type: Internal
 - Occurred date: 18/09/2025 13:43
- Right Panel:
 - General: Observables (0), TTPs (0), Attachments, Similar Cases, Similar Alerts, Responders.
 - Description: Suspicious Mimikatz activity detected on the system.
 - Tags: T1003
 - Summary: Mimikatz usage detected on host: leo (Edit)
 - Comments: Type a comment... Hit "SHIFT + ENTER" for a new line.

Add Email Notification

- Drag the **Email** app onto the board.
- Connect it after VirusTotal or TheHive.
- Configure the recipient email address.
- Paste this into the Body field:

```
Mimikatz execution identified
Time: $exec.all_fields.full_log.win.eventdata.utcTime
Title: $exec.title
Host: $exec.text.win.system.computer
```

You can also include other dynamic elements by clicking the **+** icon → **Runtime Argument** → and selecting the desired field (for example, *rule_id*, *hashes*, or *description*).



Save and run the workflow again you should now receive an email alert.



You've successfully built and automated an end-to-end SOC response pipeline. Here's what happens when **Mimikatz** is executed on your Windows VM:

1. **Wazuh** detects the activity using custom rule **100002**.
2. **Shuffle** receives the alert through the Wazuh webhook.
3. The workflow automatically **extracts the SHA256 hash** of the detected file.
4. **VirusTotal** is queried to check the file's reputation.
5. If confirmed suspicious, a **case is created in TheHive** for investigation.

6. **Email notifications** are sent to inform the analyst of the incident.

Conclusion

By completing this lab, you have successfully built a functional SOC automation pipeline that integrates Wazuh, TheHive, Shuffle, and VirusTotal.

You now understand how alerts can be automatically detected, analyzed, enriched, and escalated without manual effort.

You can extend this lab by:

- Adding more detection rules, such as PowerShell abuse or persistence techniques
- Integrating additional threat intelligence sources like AbuseIPDB or AlienVault OTX
- Creating advanced workflows in Shuffle to automate alert enrichment and case prioritization

This lab serves as a foundation for building enterprise-grade SOC automation using open-source tools. Continue exploring, testing, and improving to enhance your incident response capabilities.