
Threat Fusion: Wazuh + TheHive SOAR Integration



PROJECT REPORTED BY:

VISHWA PANCHOLI

PROJECT GUIDED BY:

POOJA MATHUR

ENROLMENT No: 202118100092

**DEPARTMENT OF ANIMATION ,
ITIMS & MOBILE APPLICATIONS
GUJARAT UNIVERSITY
BATCH 2021-2024**

GUJARAT UNIVERSITY

Department of Animation, ITIMS & Mobile Applications



Certificate

Enrolment No: 202118100092

This is to certify that Ms. Vishwa Pancholi student of MSc. IT IMS & CS (Integrated) Semester – 6, has duly completed her Term work for the semester ending in May 2024, in the subject of Project towards partial fulfilment of her Degree of Bachelor Program.

Date of Submission :
18/05/2024

Mentor
Mrs. Pooja Mathur

Index

Acknowledgement	4
1. Introduction	6
1.1 Introduction of Wazuh	6
1.2 Introduction of TheHive	9
1.3 Project Overview	11
2. Lab Design & System Requirements	13
3. Setting Up the Wazuh	15
3.1 Installation of Wazuh	15
3.2 Adding Agent in Wazuh.....	20
4. Setting Up the TheHive	25
4.1 Installation of TheHive	25
4.2 Organization and User Management	30
5. Telemetry Setup on Wazuh Dashboard	33
5.1 Integrating Sysmon and Mimikatz	33
5.2 Alert Detection Configuration in Wazuh	39
6. Workflow Creation	49
6.1 Handling Mimikatz Alerts: Shuffling Process.....	50
6.2 Converting Alert to Hash	54
6.3 Verifying Reputation Score via Virustotal	56
6.4 Creating Alerts in TheHive	60
6.5 Email Notification for SOC Analysts to Initiate Investigation	
64	
Conclusion	67
REFERENCES	68

Acknowledgement

I would like to express my sincere gratitude to all those who have contributed to the realization of this project. Firstly, I extend my deepest appreciation to my mentor and to my department their expertise and encouragement have been invaluable in shaping the direction and scope of this endeavour.

I am also indebted to my colleagues and peers for their collaboration and assistance. Their insights, feedback, and constructive criticism have significantly enriched the project, driving it towards greater refinement and effectiveness.

Furthermore, I would like to acknowledge the contributions of the open-source community, whose innovative tools and resources have formed the foundation of this project. Their commitment to knowledge sharing and collaboration has been instrumental in advancing the field of cybersecurity.

Last but not least, I am grateful to my family and friends for their patience, understanding, and encouragement during the course of this project. Their unwavering support has been a constant source of motivation and inspiration.

Student Name: Vishwa Pancholi

Enrollment Number: 202118100092

Preface

Welcome to the documentation of "ThreatFusion: Wazuh + TheHive SOAR Integration." This project represents an innovative approach to enhancing cybersecurity operations through the integration of Wazuh and TheHive SOAR platforms. By combining these powerful tools, the aim is to create a cohesive workflow that facilitates the detection and response to security threats in a timely and efficient manner.

The primary objective of this documentation is to provide a comprehensive overview of the project, including its objectives, methodologies, and implementation strategies. Through detailed explanations and practical examples, readers will gain insight into the intricacies of the integrated solution and learn how to leverage its capabilities to enhance their own security operations.

Whether you are a seasoned cybersecurity professional or a newcomer to the field, this documentation is designed to be accessible and informative. It serves as a valuable resource for anyone seeking to understand the intricacies of threat detection and response, as well as the integration of SOAR platforms into existing security infrastructure.

I hope that this documentation will serve as a useful guide and inspire further exploration and innovation in the field of cybersecurity.

1. Introduction

- Cybersecurity is increasingly important as threats become more sophisticated. "ThreatFusion: Wazuh + TheHive SOAR Integration" is a project that combines two powerful cybersecurity tools, Wazuh and TheHive with the Shuffler to make organizations safer online.
- In this intro, we'll explain what the project aims to achieve and why it's important. We'll also briefly touch on what Wazuh and TheHive are and how they work together to keep organizations secure.
- Let's get started on this journey to understand how this integration strengthens cybersecurity in simple terms.

1.1 Introduction of Wazuh

- Wazuh is a comprehensive open-source cybersecurity platform that provides threat detection, integrity monitoring, and security analytics capabilities. It is designed to help organizations enhance their security posture and protect against a wide range of cyber threats.
- Wazuh operates by collecting and analysing data from various sources within the IT environment, including logs, events, and configurations. This data is processed in real-time using a combination of predefined rules, machine learning algorithms, and anomaly detection techniques. When suspicious activity is detected, Wazuh generates alerts, initiates response actions, and provides detailed information for investigation by security analysts.

Architecture Model & Components :

- Wazuh follows a distributed architecture model, consisting of three main components:

- i. **WAZUH MANAGER :**
 - The central component of the Wazuh architecture, the Wazuh Manager is responsible for data collection, analysis, and management. It serves as the centralized server that receives data from Wazuh Agents, processes it, and generates alerts.
- ii. **WAZUH AGENT :**
 - Wazuh Agents are lightweight software agents installed on monitored systems, including servers, workstations, and cloud instances. These agents collect security-relevant data from the host system, such as logs, events, file integrity changes, and system configurations. They then forward this data to the Wazuh Manager for analysis.
- iii. **ELASTIC STACK :**
 - Wazuh integrates with the Elastic Stack, which includes Elasticsearch, Logstash, and Kibana (ELK stack), for data storage, processing, and visualization. Elasticsearch is used for indexing and searching security data, Logstash for data parsing and enrichment, and Kibana for data visualization and analysis.

Features :

- **Log and File Integrity Monitoring :**
Monitors changes to logs and critical files for signs of tampering, unauthorized access, or suspicious activities.
- **Intrusion Detection :**
Detects and alerts on known attack patterns, anomalies, and suspicious behaviour indicative of security threats, such as brute-force attacks, malware infections, and command and control communication.
- **Vulnerability Detection :**
Identifies security weaknesses, misconfigurations, and vulnerabilities in the IT infrastructure that could be exploited by attackers.

- **Compliance Monitoring :**
Helps ensure adherence to security standards, regulations, and internal security policies by monitoring for policy violations, configuration drift, and non-compliance.
- **Incident Response :**
Facilitates rapid response to security incidents by providing actionable alerts, response capabilities, and integration with incident response workflows.

Services :

- **Threat Detection :**
Provides real-time detection and alerting on security threats, allowing for timely response and mitigation to minimize the impact of security incidents.
- **Compliance Monitoring :**
Helps organizations maintain compliance with industry regulations, standards, and internal security policies by monitoring for violations and non-compliance.
- **Incident Response :**
Enables efficient incident response by providing detailed information, context, and response actions to security incidents, facilitating rapid containment and remediation.
- **Log Analysis :**
Analyses logs and events from various sources, including hosts, network devices, and applications, to identify patterns, trends, and anomalies indicative of security threats, vulnerabilities, or compliance issues.

- Wazuh's comprehensive set of features and services make it a valuable tool for organizations looking to enhance their cybersecurity posture, improve threat detection capabilities, and protect against evolving cyber threats.

1.2 Introduction of TheHive

- TheHive is an open-source Security Orchestration, Automation, and Response (SOAR) platform designed to facilitate incident response and threat intelligence activities within organizations. It enables security teams to effectively manage and respond to security incidents, streamline workflows, and automate response actions.
- TheHive works by aggregating and correlating security-related data from various sources, including SIEM systems, threat intelligence feeds, and incident reports. It provides a centralized platform for security analysts to triage, investigate, and respond to security incidents efficiently. TheHive's flexible architecture allows for integration with external tools and services, enabling seamless collaboration and automation of response actions.

Architecture Model & Components :

- **TheHive Server :**
The central component of TheHive architecture, TheHive Server is responsible for managing incidents, users, and case-related data. It provides a web-based interface for security analysts to interact with and perform various incident response tasks.
- **Case Management :**
TheHive's case management functionality allows security teams to organize and track security incidents throughout their lifecycle, from initial triage to final resolution. Cases can be assigned to individual analysts, annotated with relevant information, and linked to related incidents or observables.

- Alert Aggregation :
TheHive aggregates alerts and notifications from various sources, such as SIEM systems, threat intelligence feeds, and automated detection tools. It correlates these alerts to create unified incident records, providing a holistic view of security incidents and their associated context.

Features :

- Incident Triage :
Enables security teams to quickly triage and prioritize security incidents based on severity, impact, and urgency.
- Investigation Workflow :
Provides customizable workflows for conducting thorough investigations, including tasks, artifacts, and collaboration features.
- Response Automation :
Automates response actions and playbooks to facilitate rapid containment, mitigation, and remediation of security incidents.
- Evidence Management :
Allows for the collection, preservation, and analysis of digital evidence related to security incidents, ensuring chain of custody and integrity.
- Integration Framework :
Offers a rich set of integrations with external tools and services, including SIEM systems, threat intelligence platforms, and forensic analysis tools.

Services :

- Incident Management :
Facilitates centralized incident management, tracking, and reporting for security teams, providing visibility into incident status and progress.

- Collaboration :
Enables seamless collaboration and communication among security analysts, allowing for real-time information sharing and decision-making.
- Automation :
Automates repetitive tasks and response actions to improve efficiency and reduce response times, freeing up analysts to focus on higher-value activities.
- Intelligence Sharing :
Facilitates the sharing of threat intelligence and best practices among security teams, helping to enhance overall situational awareness and response capabilities.
- TheHive's comprehensive set of features and services make it a powerful tool for security operations teams looking to improve incident response capabilities, streamline workflows, and mitigate security risks effectively.

1.3 Project Overview

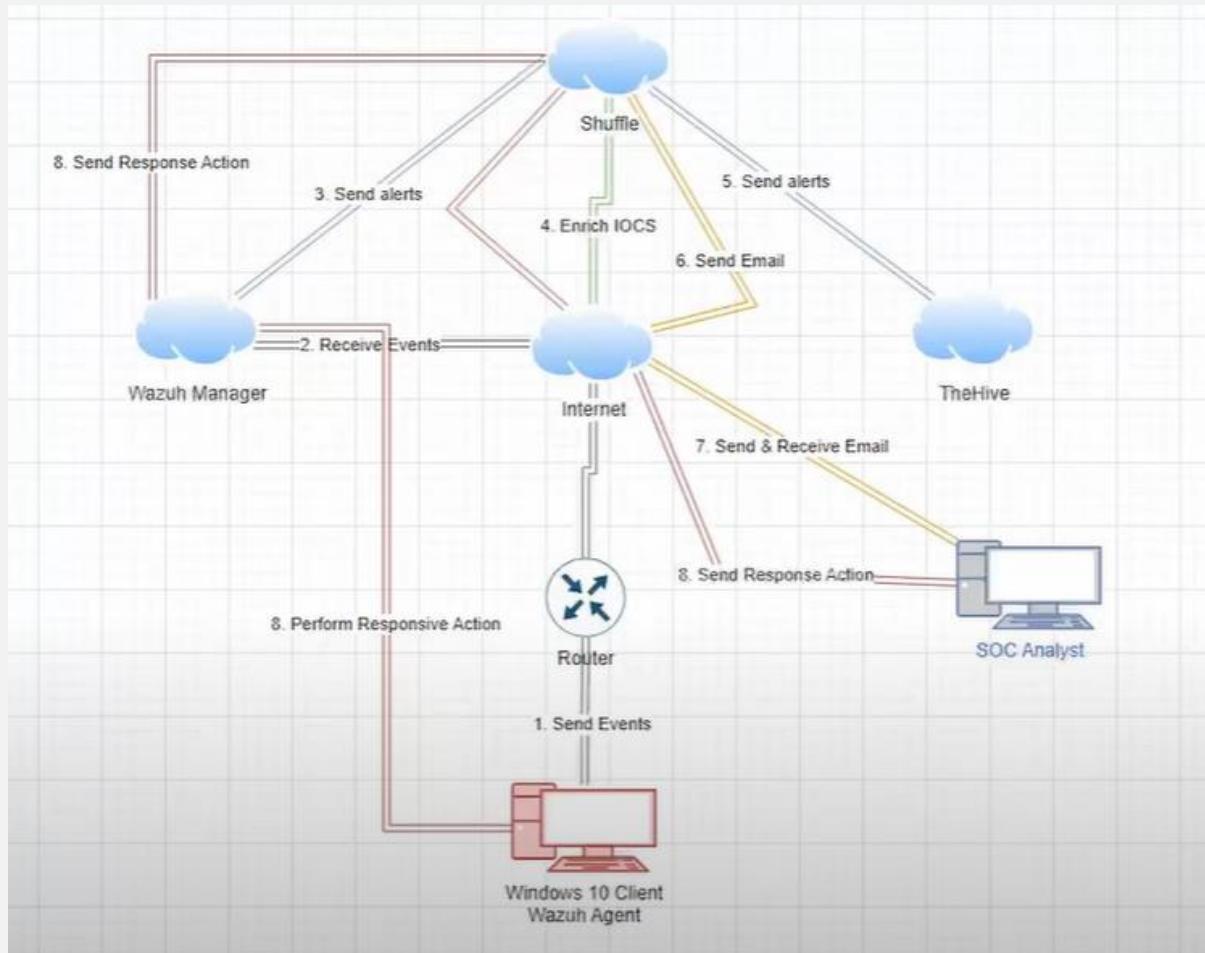
- The goal of the project "Threatfusion: Wazuh + TheHive SOAR Integration" is to create an efficient cybersecurity solution. This solution revolves around defining a practical workflow where a specific tool called Mimikatz, when used on Windows agents, is detected by the system. Once detected, an automated alert is generated and sent to a Security Operations Center (SOC) Analyst via email.
- The integration of Wazuh and TheHive SOAR holds significant importance for enhancing cybersecurity capabilities. By automating the detection and response to security threats, organizations can improve their ability to protect sensitive information, mitigate risks, and maintain operational continuity.

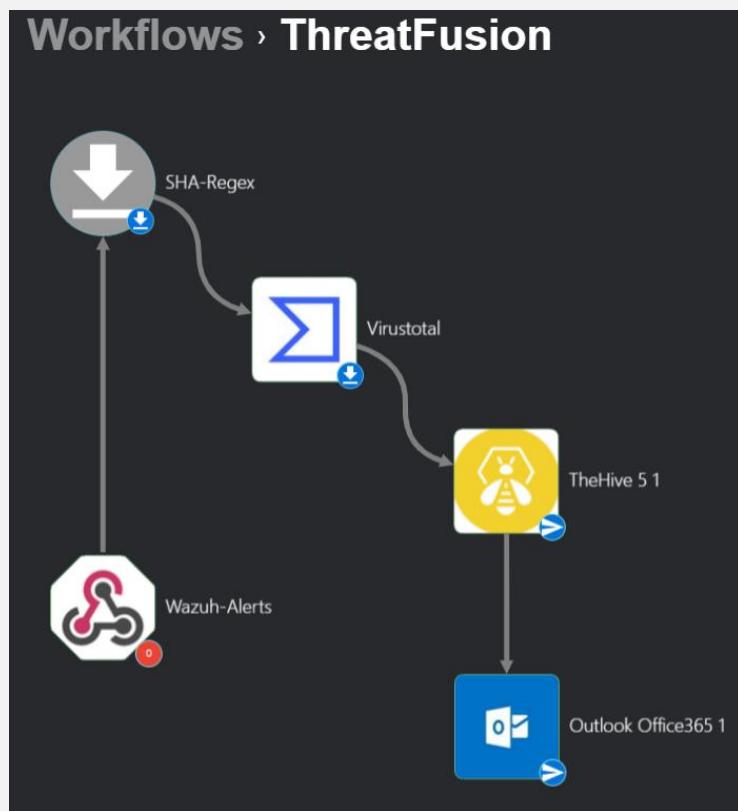
- **Integration of Wazuh and TheHive :** The project focuses on seamlessly integrating Wazuh, a security monitoring platform, with TheHive, a Security Orchestration, Automation, and Response (SOAR) platform. This integration enables the automation of incident response processes.
- **Workflow Development :** A key aspect of the project is the development of a workflow that detects the usage of Mimikatz on Windows agents. This workflow is designed to trigger automated actions, including alert generation and email notification to SOC Analysts, upon detection of suspicious activity.
- The project scope includes:
 - Configuring and deploying Wazuh and TheHive SOAR platforms.
 - Developing a custom workflow to detect the usage of Mimikatz on Windows agents.
 - Implementing automated actions, such as alert generation and email notification, upon detection of suspicious activity.
 - Testing and validating the integrated solution to ensure its effectiveness and reliability.
- This project aims to empower organizations with a proactive approach to cybersecurity, enabling them to detect and respond to security threats in a timely and efficient manner.

2. Lab Design & System Requirements

Lab Design :

- The lab design illustrates the network architecture, server setup, and integration configuration for the Threatfusion project. It provides a visual representation of how the various components interact within the lab environment.





System Requirements :

- Here's the Hardware and Software Requirements for setting up the lab environment, including server specifications, Operating Systems.
- Hardware Requirements :
 - Laptop or PC
 - RAM = 8Gb Minimum ; 16 Gb for better performance
 - Enough Storage Space

Software Requirements :

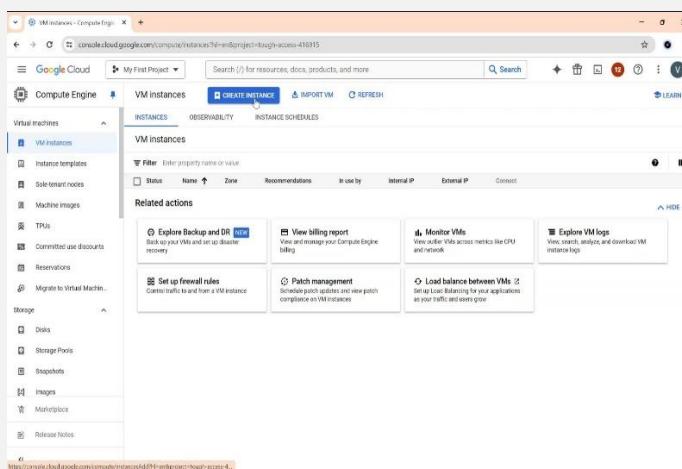
- Google Cloud Platform Free Account ; If you are developing in Cloud otherwise you can setup in your VM as well.
- VMware Workstation
- Operating System : Ubuntu or RedHat for Wazuh & TheHive and Windows 10,11 for Agent

3. Setting Up the Wazuh

- Installing Wazuh and adding agents is a crucial step in setting up the lab environment for the Threatfusion project. This section outlines the process for installing Wazuh on the central server and then adding agents to monitor target Windows systems.

3.1 Installation of Wazuh

- For the installation of Wazuh, you have the flexibility to deploy the Wazuh Manager on various platforms, including virtual machines running Ubuntu on VMware or cloud platforms like GCP, Azure, AWS. Here, we'll guide you through the process of installing Wazuh on an Ubuntu virtual machine in Google Cloud Platform(GCP).
- Next, I'll install Wazuh on Google Cloud Platform (GCP). I have a free account on GCP, and you'll need one too if you want to follow along. If you need to create an account, make sure you have access to a free account on GCP as well.



This is our GCP Console View. Here I'm in Compute Engine -> VM Instances.

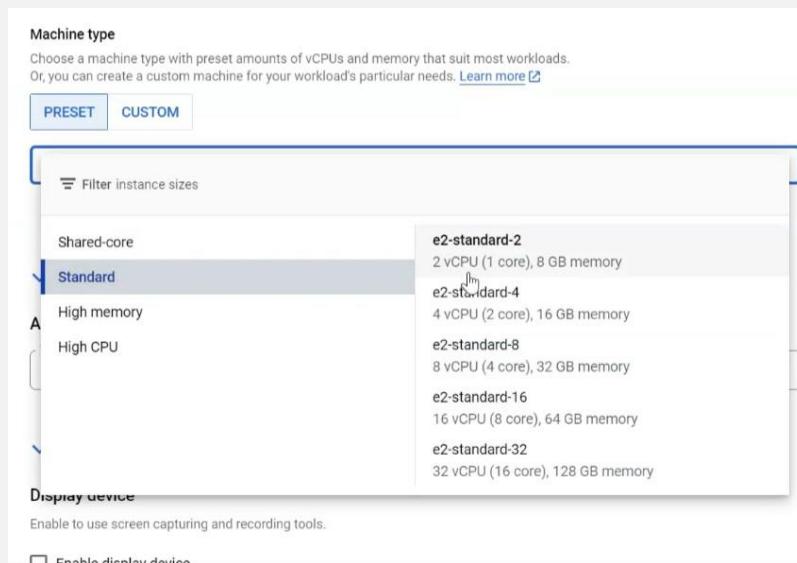
- Now we are going to follow below steps to Create the VM for our Wazuh Manager.

- Step 1:
 - We are going to Make VM for that Click on Create Instance.
 - Now, Here We are giving name to our VM as a *wazuhmanager*. and also, we are selecting our Region *Asia-south*.

The screenshot shows the 'Basic information' section of the AWS Lambda 'Create Function' wizard. It includes fields for 'Name' (wazuhmanager), 'Region' (asia-south1 (Mumbai)), 'Zone' (asia-south1-c), and a 'Machine configuration' section. Under 'Machine configuration', the 'General purpose' tab is selected, and the 'E2' series is chosen. A table lists four series: N4, C3, C3D, and E2, with E2 being the selected option.

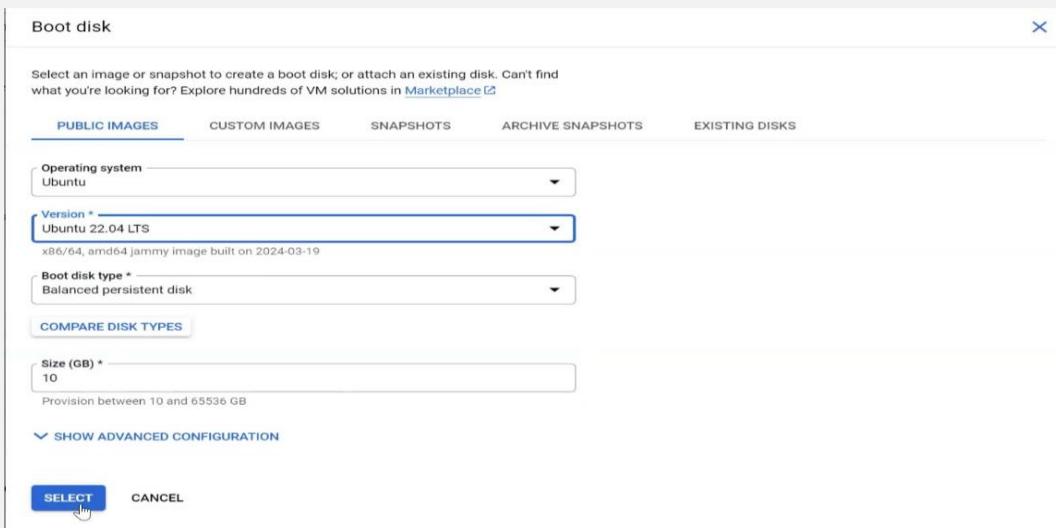
Series	Description	vCPUs	Memory	Platform
N4	Flexible & cost-optimized	2 - 80	4 - 640 GB	Intel Emerald Rapids
C3	Consistently high performance	4 - 176	8 - 1,408 GB	Intel Sapphire Rapids
C3D	Consistently high performance	4 - 360	8 - 2,880 GB	AMD Genoa
E2	Low cost, day-to-day computing	0.25 - 32	1 - 128 GB	Based on availability

- Step 2:
 - Now, we are selecting Storage; *with 2 Processor and 8 GB RAM* as this is the free account and it's enough for us.



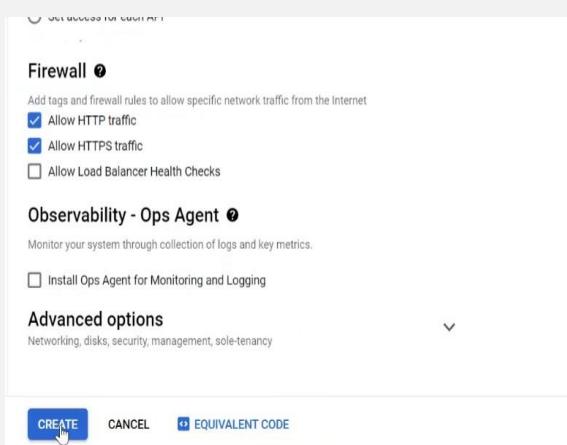
- Step 3:
 - Now, there will be an option of ENABLE DISPLAY DEVICE if you want to record and this VM so Check on that.

- Now, here you can see the option of Boot Disk Click on that from this we can customize our OS. Here, we'll select Ubuntu 22.04 latest version of OS, Disk type will be persistent disk and Disk size we're getting 10 by default but you can increase that later on.



- Step 4:

- Now, for the remote access and website access we need to allow HTTP & HTTPS traffic. So, Check on HTTP & HTTPS traffic allowance and Click on Create Option.



- After Creation of Virtual Machine, we need to modify our Firewall default rules as well. So, for that go to the Firewall.
- In the Firewall rules, edit default-allow-http rule , In the section of ports and protocols select the specific ports and Add 80 & 9000 for http.

<input type="checkbox"/>	default-allow-http	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:80,9000	Allow	1000	default	Off
<input type="checkbox"/>	default-allow-https	Ingress	https-	IP ranges: 0.0.0.0/0	tcp:443,1514-1515	Allow	1000	default	Off
<input type="checkbox"/>	inbound	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:1514-1515	Allow	1000	default	Off

- In the rule of default-allow-https add 443 & 1514-1515 for https.
- Step 5:
 - Here, we can see we successfully created our VM for WazuhManager.

The screenshot shows the GCP Compute Engine interface. At the top, there are tabs for 'VM instances', 'CREATE INSTANCE', 'IMPORT VM', and 'REFRESH'. Below the tabs, there are three navigation links: 'INSTANCES', 'OBSERVABILITY', and 'INSTANCE SCHEDULES'. The main area is titled 'VM instances' and contains a table with one row. The row has columns for 'Status' (green checkmark), 'Name' (wazuhmanager), 'Zone' (asia-south1-c), 'Recommendations', 'In use by', 'Internal IP' (10.160.0.17 (nic0)), 'External IP' (34.93.246.246 (nic0)), and 'Connect' (SSH button). Below the table, there's a section titled 'Related actions' with several buttons: 'Explore Backup and DR', 'View billing report', 'Monitor VMs', 'Explore VM logs', 'Set up firewall rules', 'Patch management', and 'Load balance between VMs'.

- Now, follow the below steps for installation of Wazuh in the VM.
- Step 1:
 - Firstly, Take the SSH access of your VM.
 - After getting SSH you'll see there's a name of your GCP username and @VM name from that switch to the root user using command : sudo su -
 - Now, Update all the existing packages and install the newer version of that using the apt-get update & upgrade command.

The screenshot shows an SSH session in a browser window titled 'SSH-in-browser'. The session URL is 'ssh.cloud.google.com/v2/ssh/projects/tough-access-416915/zones/asia-south1-c'. The terminal window shows the command 'root@wazuhmanager:~# apt-get update -y' being run. The output of the command is visible, showing 'Reading package lists... Done' and 'Building dependency tree... 50%'.

- Now, we are installing the Wazuh using the website URL which we get from the Wazuh official website documentation.

```
curl -sO https://packages.wazuh.com/4.7/wazuh-install.sh && sudo
bash ./wazuh-install.sh -a
```

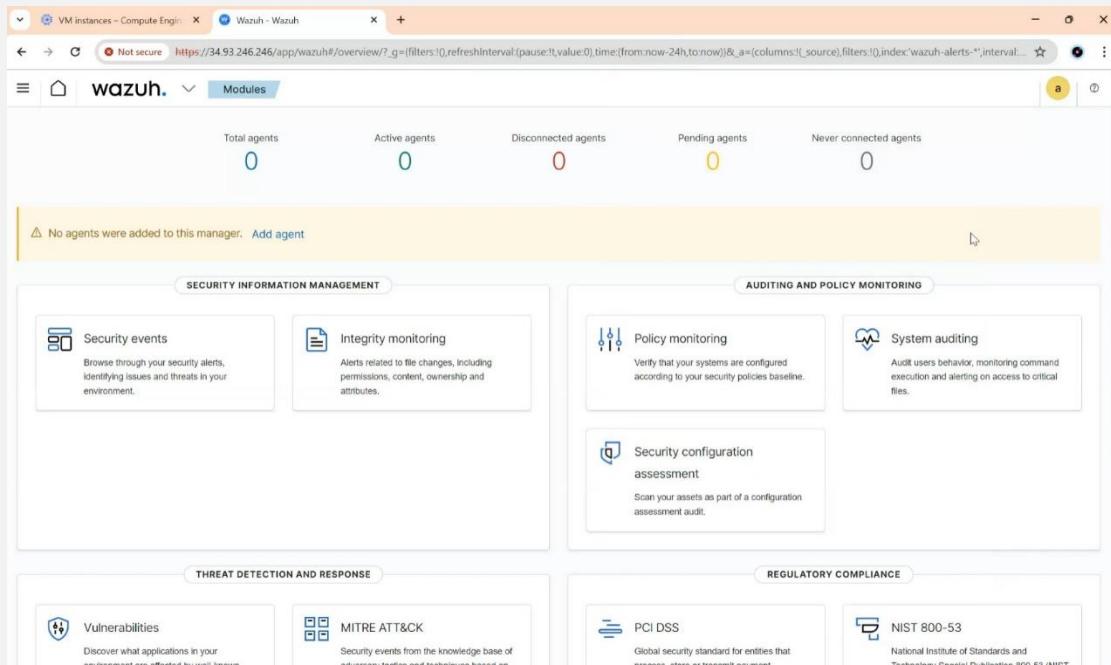
```
ssh.cloud.google.com/v2/ssh/projects/tough-access-416915/zones/asia-south1-c/instances/wazuhmanager?authuser=0&hl=en_US&projectNumber=723484402875&useAdaptiveAuth=true
ssh.cloud.google.com/v2/ssh/projects/tough-access-416915/zones/asia-south1-c/instances/wazuhmanager?authuser=0&hl=en_US&projectNumber=723484402875&useAdaptiveAuth=true
SSH-in-browser
root@wazuhmanager:~# curl -sO https://packages.wazuh.com/4.7/wazuh-install.sh && sudo bash ./wazuh-install.sh -a
```

- You can see here our installation begin and first it will install dependencies, Configuration files and then Wazuh Indexer, Wazuh Server and Wazuh Dashboard and the summary of it.

```
06:24:38 INFO: Wazuh is logging configuration to /var/log/wazuh
06:24:43 INFO: --- Dependencies ---
06:24:43 INFO: Installing apt-transport-https.
06:24:43 INFO: Wazuh repository added.
06:24:49 INFO: --- Configuration files ---
06:24:49 INFO: Generating cluster configuration files.
06:24:51 INFO: Created wazuh-install-files.tar. It contains the Wazuh cluster key, certificates, and passwords necessary for installation.
06:24:52 INFO: --- Wazuh indexer ---
06:24:52 INFO: Starting Wazuh indexer installation.
06:26:08 INFO: Wazuh indexer installation finished.
06:26:09 INFO: Wazuh indexer post-install configuration finished.
06:26:09 INFO: Starting service wazuh-indexer.
06:26:33 INFO: wazuh-indexer service started.
06:26:33 INFO: Initializing Wazuh indexer cluster security settings.
06:26:44 INFO: Wazuh indexer cluster initialized.
06:26:44 INFO: --- Wazuh server ---
06:26:44 INFO: Starting the Wazuh manager installation.
06:27:58 INFO: Wazuh manager installation finished.
06:27:58 INFO: Starting service wazuh-manager.
06:28:19 INFO: wazuh-manager service started.
06:28:19 INFO: Starting Filebeat installation.
06:28:29 INFO: Filebeat installation finished.
06:28:30 INFO: Filebeat post-install configuration finished.
06:28:30 INFO: Starting service filebeat.
06:28:31 INFO: filebeat service started.
06:28:31 INFO: --- Wazuh dashboard ---
06:28:31 INFO: Starting Wazuh dashboard installation.
06:29:34 INFO: Wazuh dashboard installation finished.
06:29:34 INFO: Wazuh dashboard post-install configuration finished.
06:29:34 INFO: Starting service wazuh-dashboard.
06:29:35 INFO: wazuh-dashboard service started.
06:30:18 INFO: Initializing Wazuh dashboard web application.
06:30:19 INFO: Wazuh dashboard web application initialized.
```

After the Complete Installation you can see there's a username and password.

- Now, Copy that username password. Now go to website search for <https://YourwazuhmanagerPublicIPAddress:443> and you'll get security error click on Advance there'll be a login page of Wazuh and paste it here that Username password.
- After a while you can see the Wazuh Dashboard that means you successfully installed Wazuh on your system.



3.2 Adding Agent in Wazuh

- Once the Wazuh manager is installed and operational, proceed to add agents to monitor target Windows systems. Configure the agent to communicate with the Wazuh manager by specifying its IP address or hostname. Verify the agent connectivity by checking the Wazuh manager console for agent registration and status.
- Before adding the agent, make sure you have a Windows 10 virtual machine (VM) installed, either on your local machine, another host, or a cloud platform, with internet connectivity, and assign it an IP address.
- Once Windows 10 is installed and connected to the internet, proceed with the following steps:
 - Step 1:
 - Now, After the Successful installation of Wazuh we see the dashboard view go to that Wazuh and at there you'll see Add Agent Option select that.

- Step 2:

- Now, deploy a New Agent in that you have to select the OS like which OS you're adding as an Agent here we're adding Windows so we'll select Windows.

Deploy new agent

1 Select the package to download and install on your system:

LINUX

- RPM amd64
- RPM aarch64
- DEB amd64
- DEB aarch64

WINDOWS

- MSI 32/64 bits

macOS

- Intel
- Apple silicon

For additional systems and architectures, please check our documentation [here](#).

2 Server address:

This is the address the agent uses to communicate with the server. Enter an IP address or a fully qualified domain name (FDQN).

Assign a server address: [?](#)

Server address

3 Optional settings:

- Step 3:

- Now, we'll give the Server IP Address which'll be the Public IP Address of our Wazuh.

Server address:

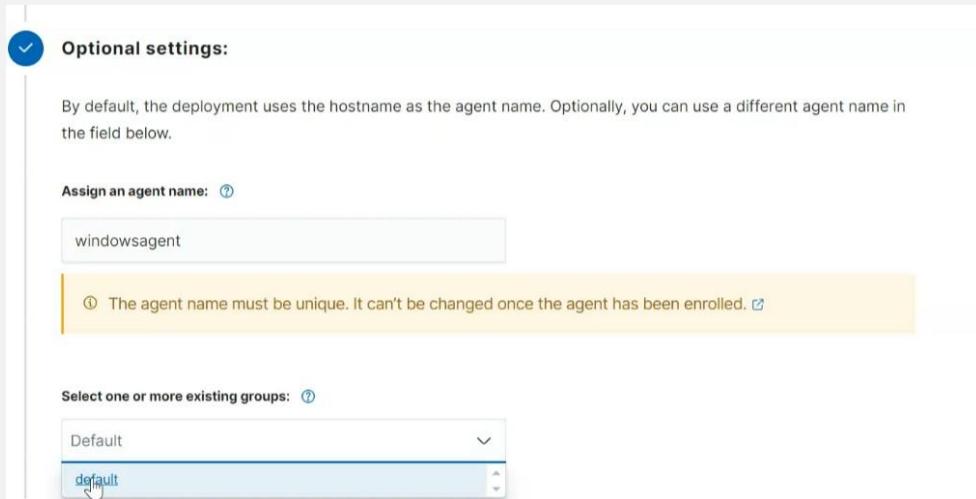
This is the address the agent uses to communicate with the server. Enter an IP address or a fully qualified domain name (FDQN.)

Assign a server address: [?](#)

34.100.231.161

- Step 4:

- After giving the IP, we'll assign a name of our agent here I'm giving windowsagent. I'm adding this agent in existing agent group which is default.



- Step 5:

- Now, run the following command in your agent. Start Windows PowerShell as an administrator and copy paste this command. It will authenticate the agent and check the connectivity.



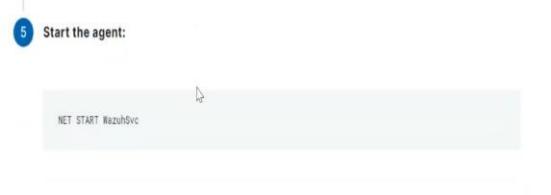
```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.7.3-1.msi -OutFile $env:tmp\wazuh-agent; msieexec.exe /i $env:tmp\wazuh-agent /q WAZUH_MANAGER='34.100.231.161' WAZUH_AGENT_GROUP='default' WAZUH_AGENT_NAME='windowsagent' WAZUH_REGISTRATION_SERVER='34.100.231.161'
PS C:\Windows\system32>
```

- Step 6:

- Once you run that command after that you need to start the Wazuh Service for that you need to run this command : NET START WazuhSvc



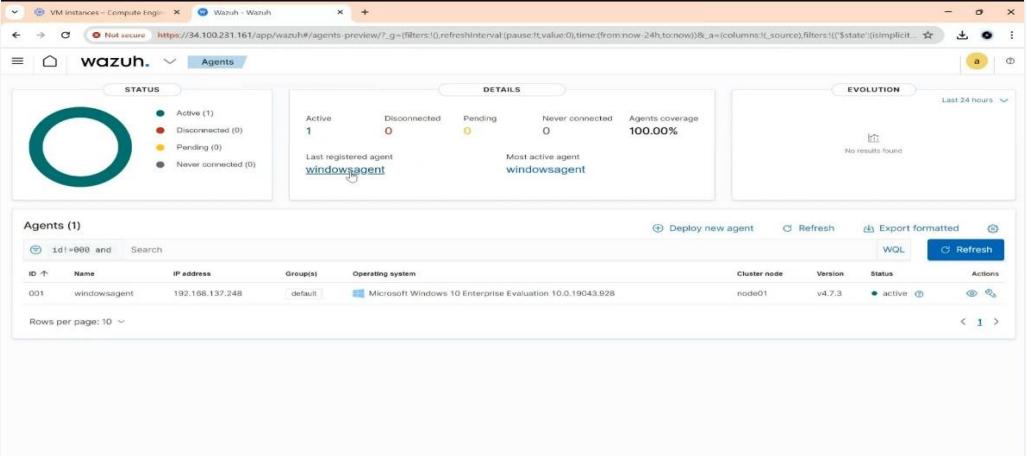
```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

PS C:\Windows\system32> Invoke-WebRequest -Uri https://(env:tmp)\wazuh-agent; msieexec.exe /i ${env:tmp}\wazuh
t' WAZUH_AGENT_NAME='windowsagent' WAZUH_REGISTRATION_
PS C:\Windows\system32> NET START WazuhSvc
The Wazuh service is starting.
The Wazuh service was started successfully.

PS C:\Windows\system32>
```

- After the Successfully running the commands in Windows 10 machine which is your Agent now. To see that agent in Wazuh Refresh or Reload the Wazuh and then you can see there's an active agent named windowsagent.



The screenshot shows the Wazuh web interface with the URL <https://34.100.231.161/app/wazuh#/agents>. The page displays the following information:

- STATUS:** Active (1), Disconnected (0), Pending (0), Never connected (0).
- DETAILS:** Active: 1, Disconnected: 0, Pending: 0, Never connected: 0, Agents coverage: 100.00%.
- EVOLUTION:** Last registered agent: [windowsagent](#), Most active agent: [windowsagent](#).
- Agents (1):**

ID	Name	IP address	Group(s)	Operating system	Cluster node	Version	Status	Actions
001	windowsagent	192.168.137.248	default	Microsoft Windows 10 Enterprise Evaluation 10.0.19043.928	node01	v4.7.3	● active	Edit Logs

- Step 7:
- Now if you want to check this through the command line or else you want to add agent through CLI Based then take SSH of wazuhmanager and go to the /var/ossec/bin/ at there you'll find the manage-agents file go into that file and from here you can edit and modify your agent.

```
root@wazuhmanager:~# cd /var/ossec/
.ssh/           api/          etc/          lib/          ruleset/        var/
active-response/ backup/      framework/    logs/          stats/         wodles/
agentless/      bin/          integrations/ queue/        tmp/
root@wazuhmanager:~# cd /var/ossec/bin/
root@wazuhmanager:/var/ossec/bin# ls
agent_control  manage_agents   wazuh-apid     wazuh-db       wazuh-logtest   wazuh-regex
agent_groups   rbac_control   wazuh-authd   wazuh-dbdb     wazuh-logtest-legacy  wazuh-remoted
agent_upgrade  verify-agent-conf wazuh-clusterd wazuh-execd  wazuh-maild    wazuh-reportd
clear_stats    wazuh-agentlessd wazuh-control  wazuh-integratord wazuh-modulesd  wazuh-syscheckd
cluster_control wazuh-analysisd wazuh-csyslogd wazuh-logcollector wazuh-monitord
root@wazuhmanager:/var/ossec/bin# ./manage_agents
```

- Here you can see it's showing our windowsagent.

```
root@wazuhmanager:/var/ossec/bin# ./manage_agents

*****
* Wazuh v4.7.3 Agent manager. *
* The following options are available: *
*****
(A)dd an agent (A).
(E)xtract key for an agent (E).
(L)ist already added agents (L).
(R)emove an agent (R).
(Q)uit.
Choose your action: A,E,L,R or Q: L

Available agents:
[1] ID: 001, Name: windowsagent, IP: any

** Press ENTER to return to the main menu.
```

4. Setting Up the TheHive

- TheHive is a powerful Security Orchestration, Automation, and Response (SOAR) platform designed to facilitate incident response and threat intelligence activities within organizations. It serves as a centralized hub for managing security incidents, enabling security teams to streamline workflows, automate response actions, and collaborate effectively.
- Now, let's proceed with the installation of TheHive on Google Cloud Platform (GCP).

4.1 Installation of TheHive

- Step 1:
 - For TheHive installation, we'll create another VM, similar to how we did for Wazuh, this time on Google Cloud Platform (GCP) as I'm performing its installation on GCP.

INSTANCES	OBSERVABILITY	INSTANCE SCHEDULES
VM instances		
<input type="button" value="Filter"/> Enter property name or value		
<input type="checkbox"/> Status	Name	Zone
<input type="checkbox"/>	thehive	asia-south1-c
<input type="checkbox"/>	wazuhmanager	asia-south1-c
Recommendations	In use by	Internal IP
		10.160.0.18 (nic0)
		34.93.246.246 (nic0)
		SSH
		⋮
External IP		
		10.160.0.17 (nic0)
		34.100.247.253 (nic0)
Connect		
		SSH
		⋮

- Firstly, Now took the SSH of TheHive and Update the existing packages and Indexes for that do apt-get update | apt-upgrade -y.

```
ssh.cloud.google.com/v2/ssh/projects/tough-access-416915/zones/asia-south1-c/instances/thehive
ssh.cloud.google.com/v2/ssh/projects/tough-access-416915/zones/asia-south1-c/instances/thehive
SSH-in-browser
root@thehive:~# apt-get update -y
0% [Working]
Get:34 http://security.ubuntu.com/ubuntu ...
Get:35 http://security.ubuntu.com/ubuntu ...
Get:36 http://security.ubuntu.com/ubuntu ...
Get:37 http://security.ubuntu.com/ubuntu ...
Get:38 http://security.ubuntu.com/ubuntu ...
Fetched 31.1 MB in 7s (4668 kB/s)
Reading package lists... Done
root@thehive:~# apt-get upgrade -y
```

- Step 2:

- Now, I'm installing TheHive from the GitHub source which I'll add at the end in the references.
- Here, we're installing TheHive using docker so for that we need to install docker in our VM so do snap install docker and apt install docker-compose.

```
root@thehive:~/docker-compose# snap install doc
docfetcher          docker          docker-image-save      doccision           doctl
root@thehive:~/docker-compose# snap install docker
[REDACTED]

[REDACTED]

doc-debian          doc-rfc-informational  docbook-dsssl      docbook-to-man    docbook-xsl-ns      dock
doc-linux-fr-html   doc-rfc-misc        docbook-dsssl-doc  docbook-utils     docbook-xsl-saxon  dock
doc-linux-fr-pdf    doc-rfc-old-std     docbook-ebnf       docbook-website   docbook2x          dock
doc-linux-fr-ps     doc-rfc-others     docbook-html-forms docbook-xml       docbook5-xml       dock
doc-linux-fr-text   doc-rfc-std       docbook-mathml     docbook-xsl       docdiff            dock
doc-rfc             doc-rfc-std-proposed docbook-simple    docbook-xsl-doc-html dochelp            dock
root@thehive:~/docker-compose# ls
docker-compose.yml
root@thehive:~/docker-compose# cd
root@thehive:~# apt install docker-compose
```

- Step 3:

- After installing docker and docker-compose we'll create a folder name docker-compose and under that we'll create a nano file named docker-compose.yml

```
root@thehive:~# ls
snap
root@thehive:~# mkdir docker-compose
root@thehive:~# ls
docker-compose  snap
root@thehive:~# cd docker-compose/
root@thehive:~/docker-compose# ls
root@thehive:~/docker-compose#
```

```
root@thehive:~# mkdir docker-compose
root@thehive:~# ls
docker-compose  snap
root@thehive:~# cd docker-compose/
root@thehive:~/docker-compose# nano docker-compose.yml
root@thehive:~/docker-compose# nano docker-compose.yml
root@thehive:~/docker-compose# sudo docker-compose up
sudo: docker-compose: command not found
root@thehive:~/docker-compose# nano docker-compose.yml
root@thehive:~/docker-compose#
```

- In that we'll add all the dependencies and also the repositories which we use for the installation of TheHive.

○ Step 4:

- We created a nano file named docker-compose.yml Now, we'll start the installation of TheHive. For that we'll run the command *docker-compose up*. Here, you can see the process has been started.

```
root@thehive:~# cd docker-compose/
root@thehive:~/docker-compose# ls
docker-compose.yml
root@thehive:~/docker-compose# nano docker-compose.yml
root@thehive:~/docker-compose# sudo do
do                         docker           docker-credential-gcloud
do-release-upgrade          docker-compose   docker-init
root@thehive:~/docker-compose# sudo docker-compose up
```

- Step 5:

- After sometime you can there were many processes are happening for Cortex, TheHive, MISP, Elastic search and many more.

```

root@thehive:~# cd docker-compose/
root@thehive:~/docker-compose# ls
docker-compose.yml
root@thehive:~/docker-compose# nano docker-compose.yml
root@thehive:~/docker-compose# sudo do
do -release-upgrade docker               docker-credential-gcloud docker-proxy
do -release-upgrade docker-compose      docker-init           docker.compose
root@thehive:~/docker-compose# sudo docker-compose up
Creating network "docker-compose_soc_Net" with driver "bridge"
Creating volume "docker-compose_miniodata" with default driver
Creating volume "docker-compose_cassandra" with default driver
Creating volume "docker-compose_elasticsearch" with default driver
Creating volume "docker-compose_thelivivedata" with default driver
Creating volume "docker-compose_mispqldata" with default driver
Creating volume "docker-compose_cassandra" with default driver
Pulling cassandra (cassandra:4)...

```

```

cortex.local_1 |      at org.apache.http.impl.nio.reactor.DefaultConnectingIOReactor.processEvent(DefaultConnectingIOReactor.java:174)
cortex.local_1 |      ... 5 common frames omitted
cortex.local_1 | INFO [main] 2024-04-26 12:25:49,547 Gossipiper.java:2293 - Waiting for gossip to settle...
cortex.local_1 | INFO [p-c-s AkkaHttpServer] 2024-04-26 12:25:49,547 Gossipiper.java:2293 - Starting Akka HTTP server...
cortex.local_1 | INFO [p-c-s AkkaHttpServer] 2024-04-26 12:25:49,547 Gossipiper.java:2293 - Listening for TCP connections on 0.0.0.0:9042
thehive_1 | [info] c.d.o.d.i.c.t.Clock [] Using native clock for microsecond precision
thehive_1 | [INFO] [main] 2024-04-26 12:25:57,549 Gossipiper.java:2234 - No gossip backlog proceeding
cassandra_1 | [INFO] [main] 2024-04-26 12:25:57,551 Gossipiper.java:2293 - Waiting for gossip to settle...
thehive_1 | [warn] c.d.o.d.i.c.c.ControlConnection [] [JanusGraph Session] Error connecting to Node(endPoint=cassandra/172.18.0.2:9042, hostId=null, hashCode=66c9ee67), trying next node (ConnectionInitException: [JanusGraph Session|control|connecting...]) Protocol initialization request, step 1 (OPTIONS): failed to send request (io.netty.channel.StacklessClosedChannelException)
thehive_1 | [warn] o.t.s.u.Retry [] An error occurs (java.lang.IllegalArgumentException: Could not instantiate implementation: org.janusgraph.diskstorage.cql.CQLStoreManager), retrying (2/10)
elasticsearch_1 | {"type": "server", "timestamp": "2024-04-26T12:26:00.266Z", "level": "INFO", "component": "o.e.x.m.p.l.CppLogMessageHandler", "cluster.name": "hive", "node.name": "03224b7cd59a", "message": "controller[100] [Main.cc@122] controller [64 bit], version 7,17,9 (Build ffceceeb3d3bc) Copyright (c) 2023 Elasticsearch BV"}
elasticsearch_1 | {"type": "server", "timestamp": "2024-04-26T12:26:01.505Z", "level": "INFO", "component": "o.e.i.g.ConfigDatabases", "cluster.name": "hive", "node.name": "03224b7cd59a", "message": "initialized default databases [[GeoLite2_Country.mmdb, GeoLite2_ASN.mmdb]], config databases [[]] and watching [/usr/share/elasticsearch/config/ingest-geoip_for changes]"}
elasticsearch_1 | {"type": "server", "timestamp": "2024-04-26T12:26:01.513Z", "level": "INFO", "component": "o.e.i.g.DatabaseNameService", "cluster.name": "hive", "node.name": "03224b7cd59a", "message": "initialized database registry, using geoip_databases directory [/tmp/elasticsearch-1706216440319356434/geoip-databases/lkHQQRWn0hOzuit8ZYCpQj]"}
elasticsearch_1 | {"type": "server", "timestamp": "2024-04-26T12:26:03.781Z", "level": "INFO", "component": "o.e.t.NettyAllocator", "cluster.name": "hive", "node.name": "03224b7cd59a", "message": "creating Elasticsearch with the following configs: [name-unpooled, suggested_max_allocation_size=1mb, factors={es.unsafe.use_unpooled_allocator=null, glibc_enabled=true, glibc_region_size=4mb, heap_size=256mb}]"}
thehive_1 | [warn] c.d.o.d.i.c.t.Clock [] Using native clock for microsecond precision
thehive_1 | [warn] c.d.o.d.i.c.c.ControlConnection [] [JanusGraph Session] Error connecting to Node(endPoint=cassandra/172.18.0.2:9042, hostId=null, hashCode=4c9297ad), trying next node (ConnectionInitException: [JanusGraph Session|control|connecting...]) Protocol initialization request, step 1 (OPTIONS): failed to send request (io.netty.channel.StacklessClosedChannelException)
elasticsearch_1 | {"type": "server", "timestamp": "2024-04-26T12:26:03.957Z", "level": "INFO", "component": "o.e.i.r.RecoverySettings", "cluster.name": "hive", "node.name": "03224b7cd59a", "message": "using rate limit [40mb] with [default=40mb, read=0b, write=0b, max=0b]"}
thehive_1 | [warn] o.t.s.u.Retry [] An error occurs (java.lang.IllegalArgumentException: Could not instantiate implementation: org.janusgraph.diskstorage.cql.CQLStoreManager), retrrying (3/10)
elasticsearch_1 | {"type": "server", "timestamp": "2024-04-26T12:26:04.161Z", "level": "INFO", "component": "o.e.d.DiscoveryModule", "cluster.name": "hive", "node.name": "03224b7cd59a", "message": "using discovery type [single-node] and seed hosts providers [{}settings]"}
thehive_1 | [INFO] [main] 2024-04-26 12:26:05,552 Gossipiper.java:2324 - No gossip backlog proceeding
cassandra_1 | [INFO] [main] 2024-04-26 12:26:05,576 TokenAllocatorFactory.java:44 - Using ReplicationAwareTokenAllocator.
cassandra_1 | [INFO] [main] 2024-04-26 12:26:05,715 TokenAllocation.java:106 - Selected tokens [414520815729581618, -3093128407685653086, 1254163775911740780, 7903047836692755215, 6152860177783069841, 538728162976407590, -39375055176814837, 436491843299389986, 2329919657137320319]
cassandra_1 | [INFO] [main] 2024-04-26 12:26:05,815 ColumnFamilyStore.java:1012 - Enqueuing flush of system_schema.columns, Reason: INTERNALLY_FORCED, Usage: 180.073KiB (0%) on-heap, 0.000KiB (0%) off-heap
elasticsearch_1 | [INFO] [PerDiskMemtableFlushWriter@0:2] 2024-04-26 12:26:05,846 Flushing.java:145 - Writing Memtable-columns@0004879978(37.780KiB serialized bytes, 250 ops, 180.073KiB (0%) on-heap, 0.000KiB (0%) off-heap), flushed range = [null, null]

```

- After some time take a new SSH but don't shutdown the existing SSH processes. In the new SSH switch to the root user and run this command here you can see our TheHive is successfully running on the port number 9000.

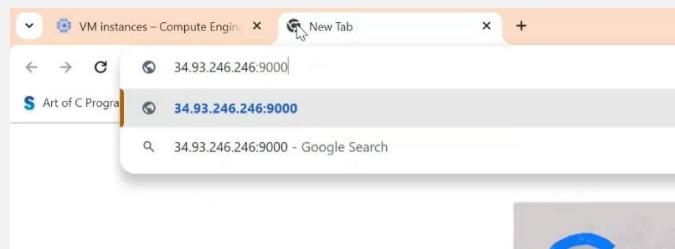
```

vishwapancholi7@thehive:~$ sudo docker ps
CONTAINER ID IMAGE NAMES COMMAND CREATED STATUS PORTS
a4d235929474 strangebee/thehive:5.2 docker-compose thehive_1 "/opt/thehive/entryp..." 3 minutes ago Up 3 minutes 0.0.0.0:9000->9000/tcp
c8c499d6d165 thehiveproject/cortex:latest docker-compose cortex.local_1 "/opt/cortex/entryp..." 3 minutes ago Up 3 minutes 0.0.0.0:9001->9001/tcp
6698f8c1e409 coolacid/misp-docker:modules-latest docker-compose misp-modules_1 "/usr/local/bin/misp..." 3 minutes ago Up 3 minutes
d4815ab210fc coolacid/misp-docker:core-latest docker-compose misp_local_1 "/entrypoint.sh" 3 minutes ago Up 3 minutes 0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp
e837fa4f81a mysql/mysql-server:5.7 docker-compose misp_mysql_1 "/entrypoint.sh mysq..." 3 minutes ago Up 3 minutes (healthy) 3306/tcp, 33060/tcp
8d0765940cad quay.io/minio/minio docker-compose minio_minio_1 "/usr/bin/docker-entry..." 3 minutes ago Up 3 minutes 9000/tcp, 0.0.0.0:9002->9002/tcp
544cbe221b19 redis:latest docker-compose redis_1 "docker-entrypoint.s..." 3 minutes ago Up 3 minutes 6379/tcp
03224b7cd59a docker.elastic.co/elasticsearch/elasticsearch:7.17.9 "/bin/tini -- /usr/l..." 3 minutes ago Up Less than a second 0.0.0.0:9200->9200/tcp, 9300/tcp
51eeb0bf8a0 cassandra:42/tcp docker-compose cassandra_1 "docker-entrypoint.s..." 3 minutes ago Up 3 minutes 7000-7001/tcp, 7199/tcp, 9160/tcp, 0.0.0.0:9042->90
vishwapancholi7@thehive:~$

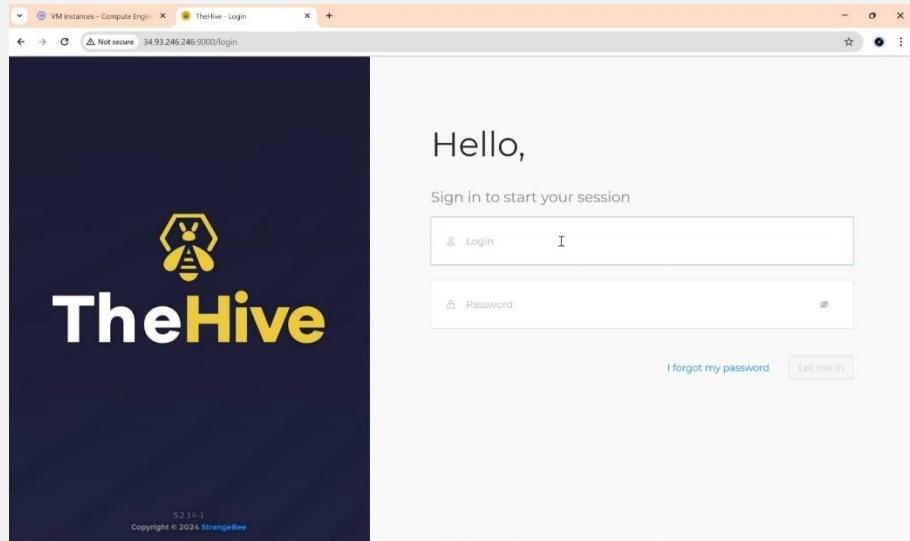
```

○ Step 6:

- Now, go to your website and copy paste your TheHive's Public IP Address with port number 9000 like this YourIPAddress:9000



- After this you'll see a login page



- Step 7:
 - Here, enters the default username password.

Username: admin@thehive.local

Password: secret

- After the logging in TheHive you can see the Console View of it.

4.2 Organization and User Management

- In this section, we'll set up user management by creating users and organizations within TheHive:
 - Step 1:
 - In your console you can see the organisation list there's a default admin organisation we'll add our own organisation named ThreatFusion.

The screenshot shows the 'Organisation List' page in TheHive. At the top, there are buttons for '+', 'default', and 'Export list'. Below is a table with columns: NAME, CREATED BY, and CREATED DATE. One organization is listed: ThreatFusion, created by 'TheHive system user' on '26/04/2024 18:01'. The organization has an 'Active' status and an 'admin' user. A note says 'Linked organisations: None'.

A modal dialog box titled 'ThreatFusion' is open. It contains fields for 'Name' (ThreatFusion) and 'Description' (Wazuh + ThreatFusion SOAR Integration). Under 'Tasks sharing rule' and 'Observables sharing rule', both are set to 'manual'. At the bottom right are 'Cancel' and 'Confirm' buttons, with 'Confirm' being highlighted.

- Step 2:
 - Now click on that organisation there's no user in it so we'll add the user in it and for Click on Add User.

The screenshot shows the 'ThreatFusion' organization page. On the left, there's a sidebar with 'Creation date' (30/04/2024 21:21), '6 seconds ago', 'Description' (Wazuh + ThreatFusion SOAR Integration), 'Tasks sharing rule' (manual), and 'Observables sharing rule' (manual). The main area has a button '+ Add User' with a cursor over it. Below it are buttons for '+', 'default', and 'Export list'. To the right, a message says 'No users have been found. Add User'.

- Here we get 2 types of Users 1st is normal and 2nd one is Service so we'll create 2 users.

1st User: Vishwa

The screenshot shows the 'Adding a User' interface for a 'Normal' user. The fields filled are:

- Type:** Normal
- Organisation:** ThreatFusion
- Login:** Vishwa@test.com
- Name:** Vishwa
- Profile:** analyst
- Permissions:** A large list of permissions including accessThehiveFS, manageAction, manageAlert/create, manageAlert/delete, manageAlert/import, manageAlert/reopen, manageAlert/update, manageAnalyse, manageCase/changeOwnership, manageCase/create, manageCase/delete, manageCase/merge, manageCase/reopen, manageCase/update, manageCaseReport, manageComment, manageCustomEvent, manageFunction/invoke, manageKnowledgeBase, manageObservable, managePage, manageProcedure, manageShare, manageTask.

At the bottom, there are 'Cancel', 'Save and add another' (highlighted with a cursor), and 'Confirm' buttons.

2nd User : SOAR

The screenshot shows the 'Adding a User' interface for a 'Service' user named SOAR. The fields filled are:

- Type:** Service
- Organisation:** ThreatFusion
- Login:** shuffle@test.com
- Name:** SOAR
- Profile:** analyst
- Permissions:** A large list of permissions identical to the first user.

At the bottom, there are 'Cancel', 'Save and add another' (highlighted with a cursor), and 'Confirm' buttons.

The screenshot shows the 'Users' page in ThreatFusion. The table lists two users:

	DETAILS	FULL NAME	LOGIN	PROFILE	MFA	DATES	C.	U.	...
5	SOAR	shuffle@test.com	shuffle	analyst		C. 03/05/2024 11:38 U. 03/05/2024 11:40			...
V	Vishwa	vishwa@test.com	vishwa	analyst		C. 03/05/2024 11:38 U. 03/05/2024 11:40			...

○ Step 3:

- Now, in the ThreatFusion User list you can see Vishwa and SOAR users. We'll edit this user by adding a password and API Key.

- We'll add Password for our normal user and for the Service user we'll generate the API Key. Now remember this API Key we'll use as in our workflow which we'll do in Shuffle.

The image displays two side-by-side screenshots of the TheHive platform's user management interface.

User Profile for Vishwa:

- Header:** Vishwa, Active
- Basic Info:** Id: ~2261040, Created by: admin@thehive.local, Created at: 03/05/2024 11:38, Updated at: 03/05/2024 11:40.
- Status:** Locked (with a toggle switch)
- MFA:** No
- API Key:** A text input field with three buttons: Create, Reveal, and Revoke.
- Profile:** Analyst
- Permissions:** A list of permissions including: accessTheHiveFS, manageAction, manageAlert/create, manageAlert/delete, manageAlert/import, manageAlert/reopen, manageAlert/update, manageAnalyse, manageCase/changeOwnership, manageCase/create, manageCase/delete, manageCase/merge, manageCase/reopen, manageCase/update, manageCaseReport, manageComment, manageCustomEvent, manageFunction/Invoke, manageKnowledgeBase, manageObservable, managePage, manageProcedure, manageShare, manageTask.
- Password:** Edit password button and a Reset the password button.
- Action Buttons:** Delete user (red button) and a large red button at the bottom.

User Profile for SOAR:

- Header:** SOAR, Active
- Basic Info:** Id: ~1192072, Created by: admin@thehive.local, Created at: 03/05/2024 11:38, Updated at: 03/05/2024 11:40.
- Profile:** SOAR (represented by a question mark icon)
- Login:** shuffle@test.com
- Email:** Email
- Type:** Service
- Status:** Locked (with a toggle switch)
- MFA:** No
- API Key:** A text input field with three buttons: Renew, Reveal, and Revoke.
- Action Buttons:** Delete user (red button) and a large red button at the bottom.

5. Telemetry Setup on Wazuh Dashboard

- In this section, we'll integrate the Sysmon tool and the Mimikatz tool into our Windows agents. Subsequently, we'll observe Mimikatz alerts within Wazuh. Here's a brief description of Sysmon and Mimikatz:
 - Sysmon is a Windows system service and device driver that monitors and logs system activity to the Windows event log. It provides detailed information about process creations, network connections, and changes to file creation time. Sysmon enhances visibility into system activity, aiding in threat detection and incident response by capturing valuable forensic data.
- Sysmon (System Monitor) :
 - Sysmon is a Windows system service and device driver that monitors and logs system activity to the Windows event log. It provides detailed information about process creations, network connections, and changes to file creation time. Sysmon enhances visibility into system activity, aiding in threat detection and incident response by capturing valuable forensic data.
- Mimikatz :
 - Mimikatz is a powerful post-exploitation tool that allows users to extract credentials from memory, perform pass-the-hash attacks, and perform other credential manipulation tasks. It is commonly used by attackers to escalate privileges and move laterally within a compromised network. Detecting Mimikatz activity is crucial for identifying potential security breaches and responding to them effectively.

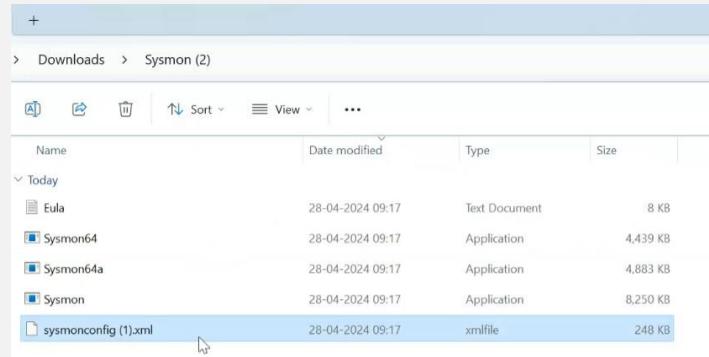
5.1 Integrating Sysmon and Mimikatz

- Now, let's proceed with adding Sysmon and Mimikatz to our Windows agents and configuring Wazuh to detect Mimikatz alerts.
 - Step 1:
 - We are generating the alert of Mimikatz in Wazuh so for that firstly we'll install the sysmon in our windows agent for that I took the GitHub source so from that i download the file of sysmon configuration and also download the sysmon tool zip from its official website and as well extract that folder and configuration file in it.

The image shows two side-by-side browser windows. The left window is the Microsoft Build website displaying the article "Sysmon v15.14". The right window is a GitHub repository page for "olafhartong/sysmon-modular", specifically the "sysmonconfig.xml" file.

○ Step 2:

- Now, we download this sysmon for that start the PowerShell as an administrator and go into that Directory where you installed folder copy paste that path now, we'll run this command : [Folder name] -i [configurationfile]

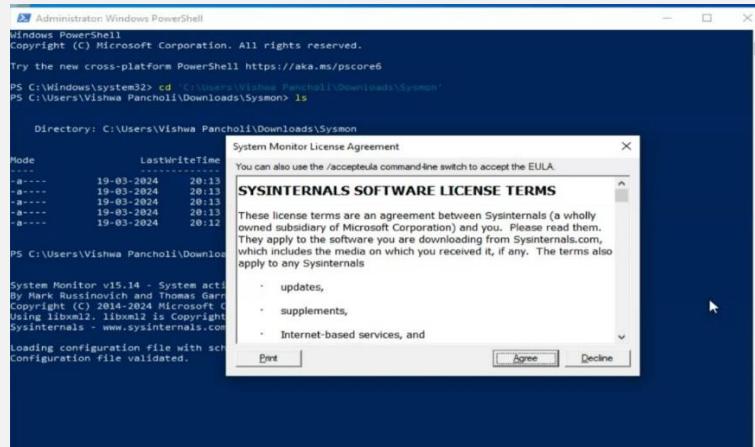


An Administrator: Windows PowerShell session. The command entered is:


```
PS C:\Windows\system32> cd 'C:\Users\Vishwa Pancholi\Downloads\Sysmon'
```

An Administrator: Windows PowerShell session. The command entered is:


```
PS C:\Users\Vishwa Pancholi\Downloads\Sysmon> .\Sysmon64.exe -i .\sysmonconfig.xml
```



- You can see there's a popup for the Agree Terms and Conditions and you successfully installed the sysmon.

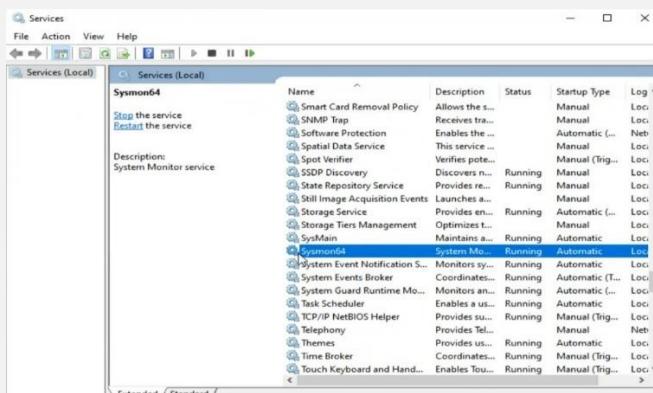
```
PS C:\Users\Vishwa Pancholi\Downloads\Sysmon> .\Sysmon64.exe -i .\sysmonconfig.xml

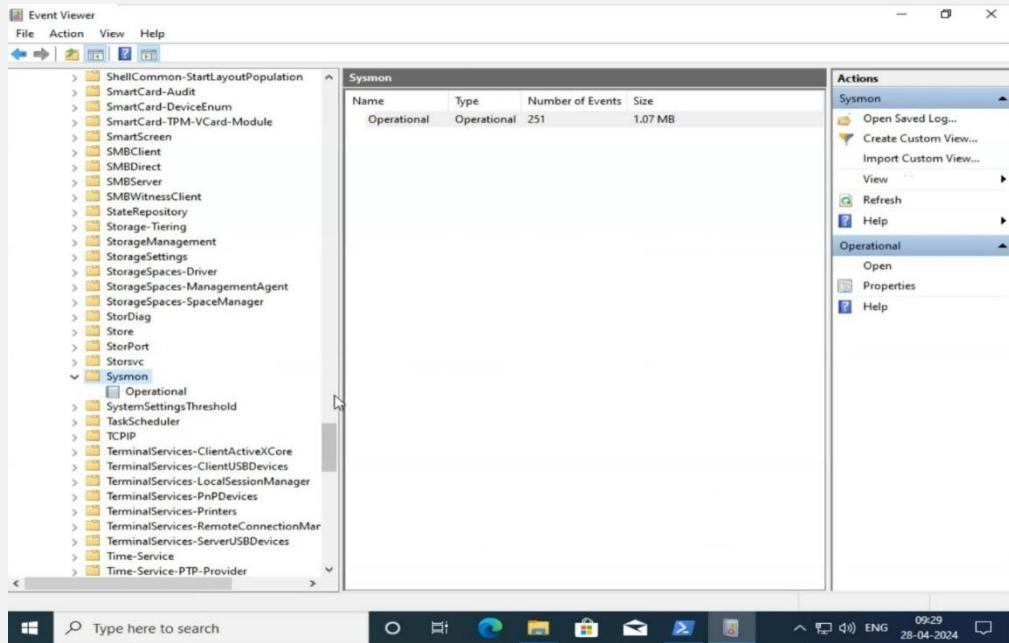
System Monitor v15.14 - System activity monitor
By Mark Russinovich and Thomas Garnier
Copyright (C) 2014-2024 Microsoft Corporation
Using libxml2. libxml2 is Copyright (C) 1998-2012 Daniel Veillard. All Rights Reserved.
Sysinternals - www.sysinternals.com

Loading configuration file with schema version 4.90
Configuration file validated.
Sysmon64 installed.
SysmonDrv installed.
Starting SysmonDrv.
SysmonDrv started.
Starting Sysmon64..
Sysmon64 started.
PS C:\Users\Vishwa Pancholi\Downloads\Sysmon>
```

○ Step 3:

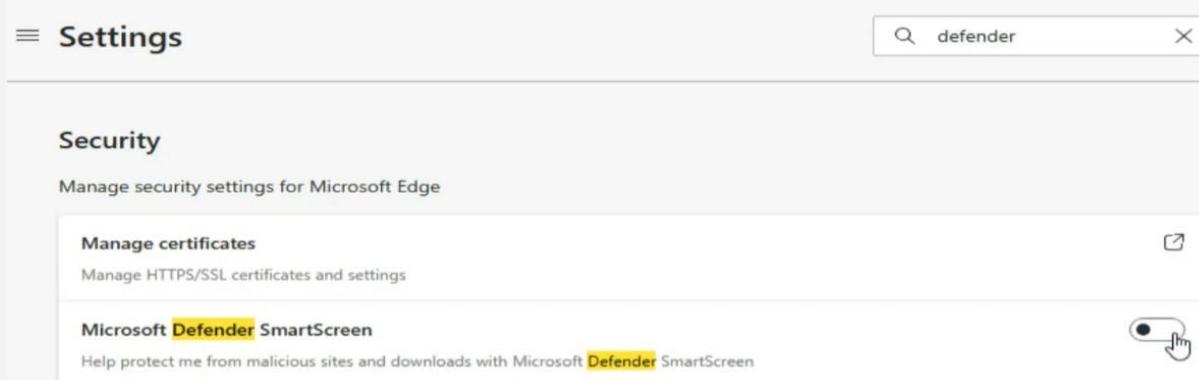
- Now you can verify this by going to services there will be a sysmon64 service or else you can go to event viewer -> Applications and Service Logs -> Microsoft -> Windows -> Sysmon





- Step 4:

- Now we are going to add Mimikatz tool in our windows agent so for that firstly we need to do some security configuration.
- Firstly, go to your browser -> settings -> Search for Defender -> Microsoft Defender SmartScreen -> disable that option.

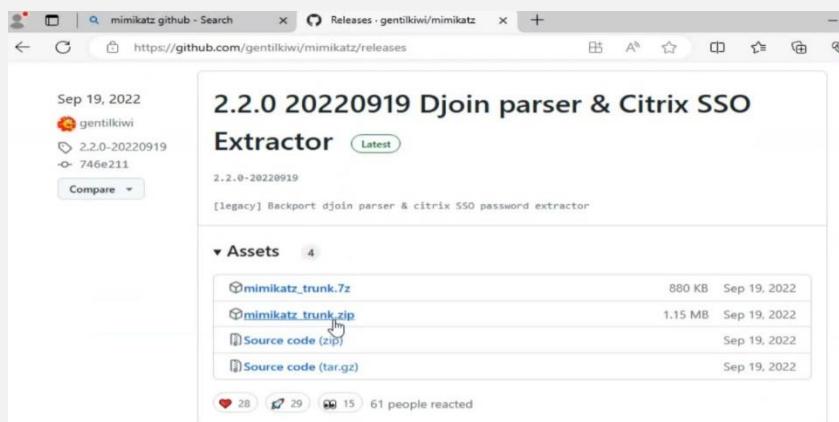


- After this Go to Windows Security -> Virus & threat Protection -> Under the Virus & threat Protection go to Manage settings -> Exclusion -> Click on Add or Remove -> Click on Add an Exclusion -> Select Folder Option and Select the downloads and done.

The screenshot shows the Microsoft Defender Antivirus interface. On the left, there's a sidebar with sections for 'Virus & threat protection' (status: Protection for your device against threats), 'Current threats' (status: No current threats, Last scan: Not available), and buttons for 'Quick scan', 'Scan options', 'Allowed threats', 'Protection history', and a cursor icon. Below this is a section for 'Virus & threat protection settings' with 'No action needed.' and a 'Manage settings' link. The main content area is titled 'Exclusions' with the sub-instruction 'Add or remove items that you want to exclude from Microsoft Defender Antivirus scans.' At the bottom, there's a large button labeled '+ Add an exclusion' and a sample entry 'C:\Users\Vishwa Pancholi\Downloads Folder'.

- Step 5:

- Now, I'm downloading the Mimikatz tool from the GitHub source I'll added that link in References and after downloading the file extract that file.



- Now, go to the Mimikatz file -> x64 and copy the path and paste it in a PowerShell and do the ls you can see all files now run that mimikatz.exe file

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

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> cd 'C:\Users\Vishwa Pancholi\Downloads\mimikatz_trunk\x64'
PS C:\Users\Vishwa Pancholi\Downloads\mimikatz_trunk\x64>
```

```

PS C:\Windows\system32> cd 'C:\Users\Vishwa_Pancholi\Downloads\mimikatz_trunk\x64'
PS C:\Users\Vishwa_Pancholi\Downloads\mimikatz_trunk\x64> ls

Directory: C:\Users\Vishwa_Pancholi\Downloads\mimikatz_trunk\x64

Mode                LastWriteTime         Length Name
----                -----          ----  --
-a---        28-04-2024     11:10           37208 mimidrv.sys
-a---        28-04-2024     11:10        1355264 mimikatz.exe
-a---        28-04-2024     11:10           37376 mimilib.dll
-a---        28-04-2024     11:10          10752 mimispool.dll

PS C:\Users\Vishwa_Pancholi\Downloads\mimikatz_trunk\x64>

```

- And you can see you enter into the Mimikatz tool.

```

PS C:\Windows\system32> cd 'C:\Users\Vishwa_Pancholi\Downloads\mimikatz_trunk\x64'
PS C:\Users\Vishwa_Pancholi\Downloads\mimikatz_trunk\x64> ls

Directory: C:\Users\Vishwa_Pancholi\Downloads\mimikatz_trunk\x64

Mode                LastWriteTime         Length Name
----                -----          ----  --
-a---        28-04-2024     11:10           37208 mimidrv.sys
-a---        28-04-2024     11:10        1355264 mimikatz.exe
-a---        28-04-2024     11:10           37376 mimilib.dll
-a---        28-04-2024     11:10          10752 mimispool.dll

PS C:\Users\Vishwa_Pancholi\Downloads\mimikatz_trunk\x64> ./mimikatz.exe

#####
# mimikatz 2.2.0 (x64) #19041 Sep 19 2022 17:44:08
## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > https://blog.gentilkiwi.com/mimikatz
## v ##> Vincent LE TOUX ( vincent.letoux@gmail.com )
#####> https://pingcastle.com / https://mysmartlogon.com ***/
mimikatz #

```

- Step 6:

- Now, for generating alerts we need to do integration of Sysmon and Mimikatz first.
- So, for that we'll go to our C drive and there will be a folder named ossec-agent. Go into that folder here you can see the configuration file named ossec.conf we're going to edit that file so before editing that file make a copy of that file named ossec-backup.conf.



- Now Open that original file in notepad as an administrator.
- Now, firstly remove the below configuration :

```

</localfile>
<location>Security</location>
<log_format>eventchannel</log_format>
<query>Event/System[EventID != 5145 and EventID != 5156 and EventID != 5447 and
EventID != 4656 and EventID != 4658 and EventID != 4663 and EventID != 4660 and
EventID != 4670 and EventID != 4690 and EventID != 4703 and EventID != 4907 and
EventID != 5152 and EventID != 5157]</query>
</localfile>

<localfile>
<location>System</location>
<log_format>eventchannel</log_format>
</localfile>

<localfile>
<location>active-response\active-responses.log</location>
<log_format>syslog</log_format>
</localfile>

<!-- Policy monitoring -->
<rootcheck>
<disabled>no</disabled>

```

- And in local file add the Sysmon Full Name from the Event Viewer and edit this as shown below and save this file.

```

</client_buffer>

<!-- Log analysis -->
<localfile>
<location>Microsoft-Windows-Sysmon/Operational</location>
<log_format>eventchannel</log_format>
</localfile>

<localfile>
<location>active-response\active-responses.log</location>
<log_format>syslog</log_format>
</localfile>

<!-- Policy monitoring -->
<rootcheck>
<windows_apps>./shared/win_applications_rcl.txt</windows_apps>
<windows_malware>./shared/win_malware_rcl.txt</windows_malware>
</rootcheck>

```

5.2 Alert Detection Configuration in Wazuh

- In this section, we'll generate alerts from the perspective of an attacker, simulating Mimikatz activity. To accomplish this, we'll need to edit rules and configuration files within Wazuh. Let's proceed with making the necessary adjustments:
 - Step 1:
- Now, go to the wazuhmanager CLI based and copy the config file as an ossec-backup.conf at a different location.

SSH-in-browser

```
root@wazuhmanager:~# cp /var/ossec/etc/ossec.conf ~/ossec-backup.conf
root@wazuhmanager:~#
```

- Now, edit the config file in that <logall>yes</logall> ;
<logall_json>yes</logall_json> and restart the wazuh-manager service

```
root@wazuhmanager:~# cp /var/ossec/etc/ossec.conf ~/ossec-backup.conf
root@wazuhmanager:~# nano /var/ossec/etc/ossec.conf
```

SSH-in-browser

```
GNU nano 6.2
<!--
  Wazuh - Manager - Default configuration for ubuntu 22.04
  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>
  </global>
```

- Step 2:

- Now, edit the file /etc/filebeat/filebeat.yml in the section of filebeat modules there's an archive enabled make it as a true and save and quit and restart the service of filebeat.

SSH-in-browser

```
root@wazuhmanager:~# cp /var/ossec/etc/ossec.conf ~/ossec-backup.conf
root@wazuhmanager:~# nano /var/ossec/etc/ossec.conf
root@wazuhmanager:~# nano /var/ossec/etc/ossec.conf
root@wazuhmanager:~# systemctl restart wazuh-manager.service
root@wazuhmanager:~# cd /var/ossec/1
lib/_logs/
root@wazuhmanager:~# cd /var/ossec/logs/
root@wazuhmanager:/var/ossec/logs# ls
active-responses.log alerts api.log archives cluster cluster.log firewall integrations.log ossec.log wazuh
root@wazuhmanager:/var/ossec/logs# cd archives/
root@wazuhmanager:/var/ossec/logs/archives# ls
2024 archives.json archives.log
root@wazuhmanager:/var/ossec/logs/archives# nano archives.
root@wazuhmanager:/var/ossec/logs/archives# nano /etc/filebeat/filebeat.yml
root@wazuhmanager:/var/ossec/logs/archives#
```

```

GNU nano 6.2
# Wazuh - Filebeat configuration File
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'
setupilm.overwrite: true
setupilm.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

```

- Step 3:
- For the Dashboard Changes go to setting -> Stack Management -> Index Patterns -> Create Index Pattern -> Enter the name : wazuh-archives-** -> Next Step -> Choose @timestamp in time field -> And create.

The screenshot shows the OpenSearch Dashboards interface with the 'Stack Management' tab selected. On the left, there's a sidebar with 'Management' and 'Stack Management' highlighted. The main area is titled 'OpenSearch Dashboards' and shows 'Index Patterns'. A blue button labeled '+ Create index pattern' is prominently displayed at the top of the index patterns list. Below it, there's a search bar and some other UI elements.

Step 1 of 2: Define an index pattern

Index pattern name: wazuh-archives-**

Use an asterisk (*) to match multiple indices. Spaces and the characters \, /, ?, *, <, >, | are not allowed.

Include system and hidden indices

✓ Your index pattern matches 1 source.

wazuh-archives-4.x-2024.04.28 Index

Rows per page: 10

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: @timestamp Refresh

Show advanced settings

wazuh-archives-** Fields (501) Scripted fields (0) Source filters (0)

Name	Type	Format	Searchable	Aggregatable	Excluded
@timestamp	date		•	•	
@version	string		•		
GeoLocation.area_code	number		•	•	
GeoLocation.city_name	string		•	•	
GeoLocation.continent_code	string		•		
GeoLocation.coordinates	number		•	•	
GeoLocation.country_code2	string		•		
GeoLocation.country_code3	string		•		
GeoLocation.country_name	string		•	•	
GeoLocation.district_code	number		•	•	

- Here You can see the New Index Pattern.
- Now, Go to Setting -> Discover -> Change Index Pattern -> Select wazuh-archives-**

Stack Management Index patterns Create index

Recently viewed: No recently viewed items

W Wazuh: Wazuh

OpenSearch Dashboards: Discover (selected), Dashboard, Visualize

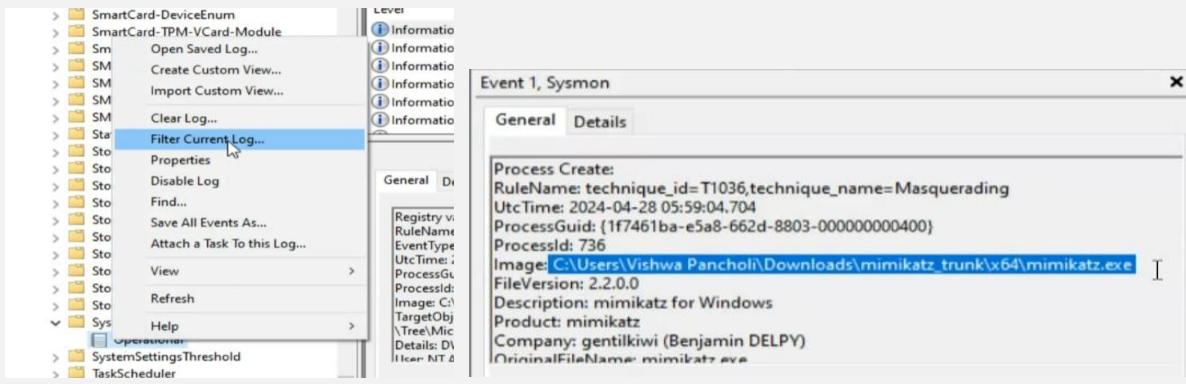
Create index pattern: Step 1 of 2: Define an index pattern

CHANGE INDEX PATTERN

Filter options: wazuh-alerts-* (selected), wazuh-archives-**, wazuh-monitoring-*, wazuh-statistics-*

agent.name Time

- Step 4:
- Now, go to your agent and run the Mimikatz.exe again and after that go to event viewer -> Applications and Service Logs -> Microsoft -> Windows -> Sysmon -> right click on Operation -> Select Filter Current Log -> and above task category write Event Id : 1 -> and in the below details you'll find mimikatz path in Image option.



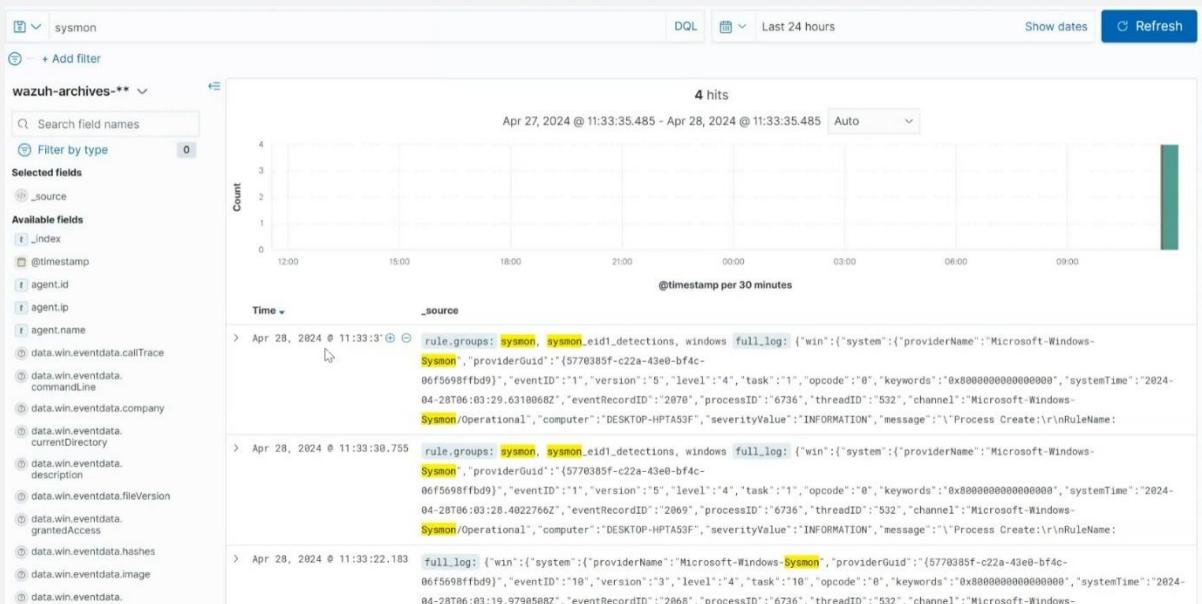
- Now, go to services and restart the wazuh service.

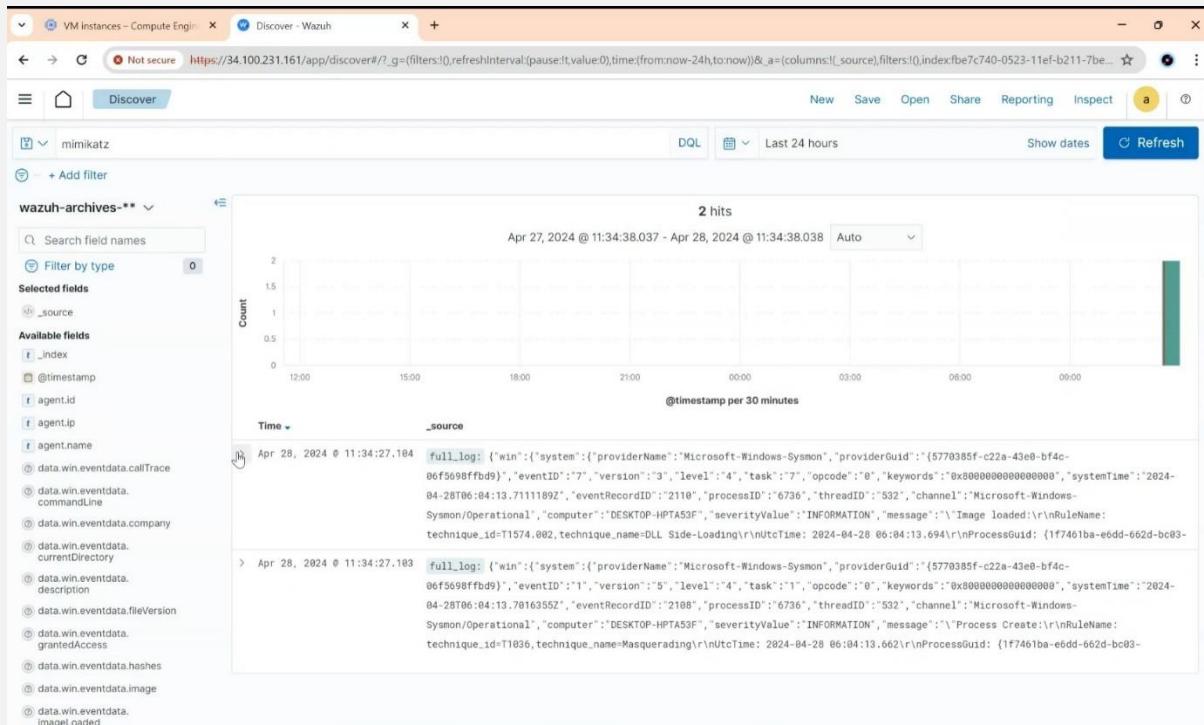
○ Step 5:

- Now, go to SSH of wazuhmanager and find any logs related to mimikatz if detected :
 - cat archives.json | grep -i mimikatz

○ Step 6:

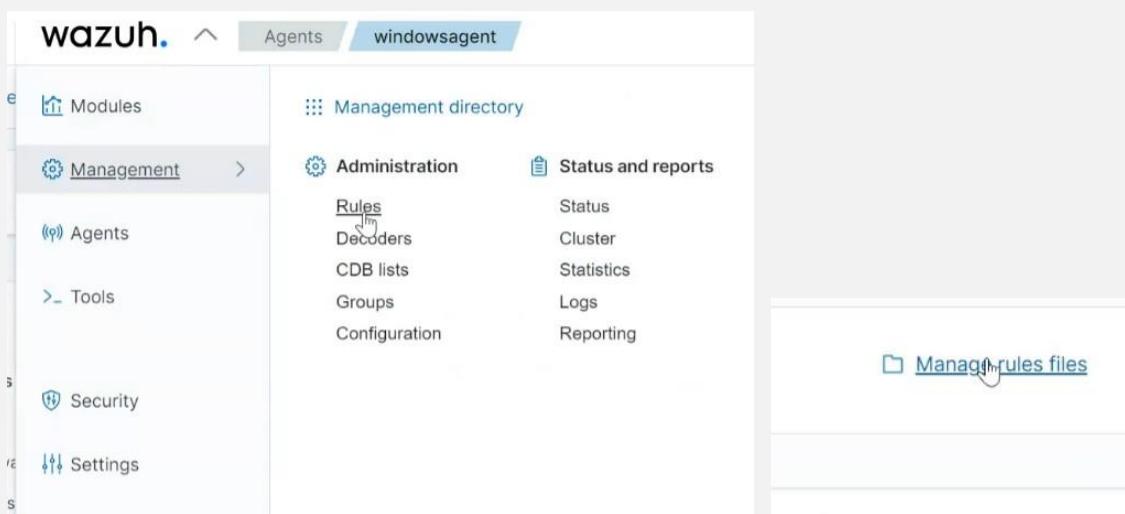
- Go to Wazuh Dashboard and at their search for the sysmon tool and mimikatz tool so, at there you'll find the logs related to that event.





○ Step 7:

- Now, we'll modify our mimikatz file still want to detect our event because you never know attacker can change the file name and do the attack so for that we'll edit our rules from the Wazuh Dashboard.
 - Go to dashboard -> Management -> Administration -> Rules -> Manage rule files -> search for sysmon rule



Rules files (10)		
From here you can manage your rules files.		
	Path	Actions
0330-sysmon_rules.xml	ruleset/rules	
0595-win-sysmon_rules.xml	ruleset/rules	
0800-sysmon_id_1.xml	ruleset/rules	
0810-sysmon_id_3.xml	ruleset/rules	
0820-sysmon_id_7.xml	ruleset/rules	
0830-sysmon_id_11.xml	ruleset/rules	
0860-sysmon_id_13.xml	ruleset/rules	
0870-sysmon_id_8.xml	ruleset/rules	
0945-sysmon_id_10.xml	ruleset/rules	
0950-sysmon_id_20.xml	ruleset/rules	

Rows per page: 10 ▾

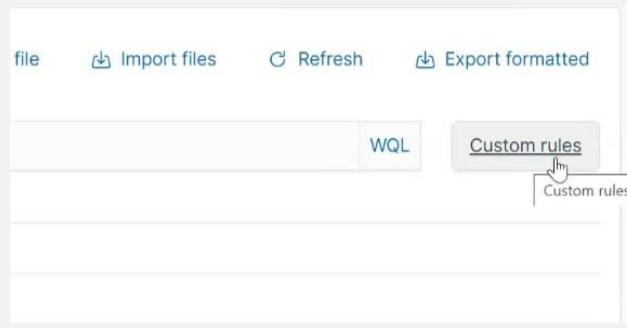
↳ 0800-sysmon_id_1.xml

```

1 * <!--
2   Copyright (C) 2015, Wazuh Inc.
3 -->
4
5 * <!--
6   Sysmon Event ID 1 rules: 92000 - 92100
7 -->
8
9 * <group name="sysmon,sysmon_eid1_detections,windows,">
10
11 *   <rule id="92000" level="4">
12     <if_group>sysmon_event1</if_group>
13     <field name="win.eventdata.parentImage" type="pcre2">(?i)\((c|w)script\|.exe</field>
14     <options>no_full_log</options>
15     <description>Scripting interpreter spawned a new process</description>
16   *   <mitre>
17     <id>T1059.005</id>
18   </mitre>
19 </rule>
20
21 *   <rule id="92001" level="6">
22     <if_sid>92000</if_sid>
23     <field name="win.eventdata.commandLine" type="pcre2">(?i)\((c|w)script\|.exe.+\.(bat|cmd|lnk|pif|vbs|vbe|js|wsh|ps1)</field>
24     <options>no_full_log</options>
25     <description>Scripting interpreter spawned new scripting interpreter</description>
26   *   <mitre>
27     <id>T1059</id>
28   </mitre>
29 </rule>
30
31 *   <rule id="92002" level="6">
32     <if_sid>92000</if_sid>
33     <field name="win.eventdata.commandLine" type="pcre2">\cmd\|.exe</field>
34     <options>no_full_log</options>
35     <description>Scripting interpreter spawned Windows command shell instance</description>
36   <mitre>

```

- It will show you a list of sysmon rule; go in any file and copy any rule; Now, go to custom rules -> Edit the local_rules.xml file -> Paste that copied rule here



Rules files (1)
From here you can manage your rules files.

File ↑	Path	
local_rules.xml	etc/rules	Edit local_rules.xml content

Rows per page: 10 ▾

- Now, modified the rule as per your need.

local_rules.xml

```

1  <!-- Local rules -->
2
3  <!-- Modify it at your will. -->
4  <!-- Copyright (C) 2015, Wazuh Inc. -->
5
6  <!-- Example -->
7+ <group name="local,syslog,sshd,">
8
9+   <!--
10    Dec 10 01:02:02 host sshd[1234]: Failed none for root from 1.1.1.1 port 1066 ssh2
11    --
12+   <rule id="100001" level="5">
13     <if_sid>5716</if_sid>
14     <srcip>1.1.1.1</srcip>
15     <description>sshd: authentication failed from IP 1.1.1.1.</description>
16     <group>authentication_failed,pci_dss_10.2.4,pci_dss_10.2.5,</group>
17   </rule>
18
19+   <rule id="100002" level="15">
20     <if_group>sysmon_event1</if_group>
21     <field name="win.eventdata.originalFileName" type="pcre2">(?i)mimikatz\.exe</field>
22     <description>Mimikatz Usage Detected</description>
23+   <mitre>
24     <id>T1003</id>
25   </mitre>
26 </rule>
27
28 </group>
29

```

- Save the editing and restart the Manager.
 - Step 8:
- Now, go to the Windows Agent and rename the mimikatz file here I named it as a Hackerspace.

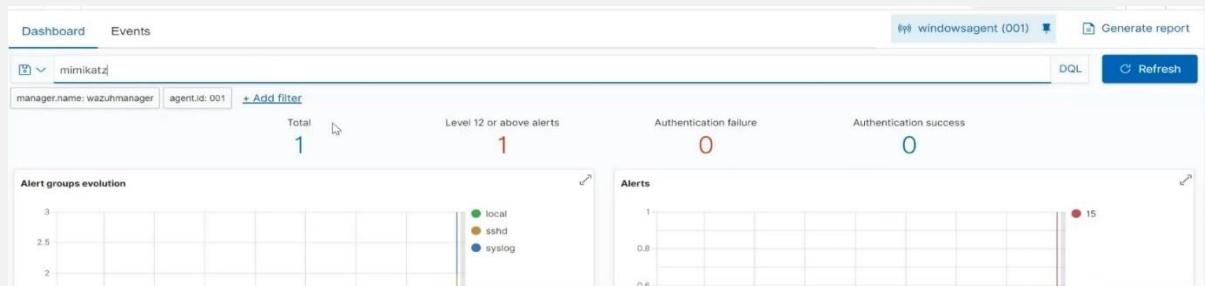
Name	Date modified	Type
Hackerspace	28-04-2024 11:10	App
mimidrv.sys	28-04-2024 11:10	Sys
mimilib.dll	28-04-2024 11:10	App
mimispool.dll	28-04-2024 11:10	App

- Now, go to PowerShell and run the hackerspace file.

```
PS C:\Users\Vishwa Pancholi\Downloads\mimikatz_trunk\x64> .\Hackerspace.exe
.#####. mimikatz 2.2.0 (x64) #19041 Sep 19 2022 17:44:08
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > https://blog.gentilkiwi.com/mimikatz
'## v ##' Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > https://pingcastle.com / https://mysmartlogon.com ***/
mimikatz #
```

○ Step 9:

- Now, go to the SSH Manager and run the cat archives.json | grep -i Hackerspace there will be a log of it and go to the Dashboard and search for mimikatz it will show 1 alert named Mimikatz Usage Detected and check originalFilename under logs it will be mimikatz.exe



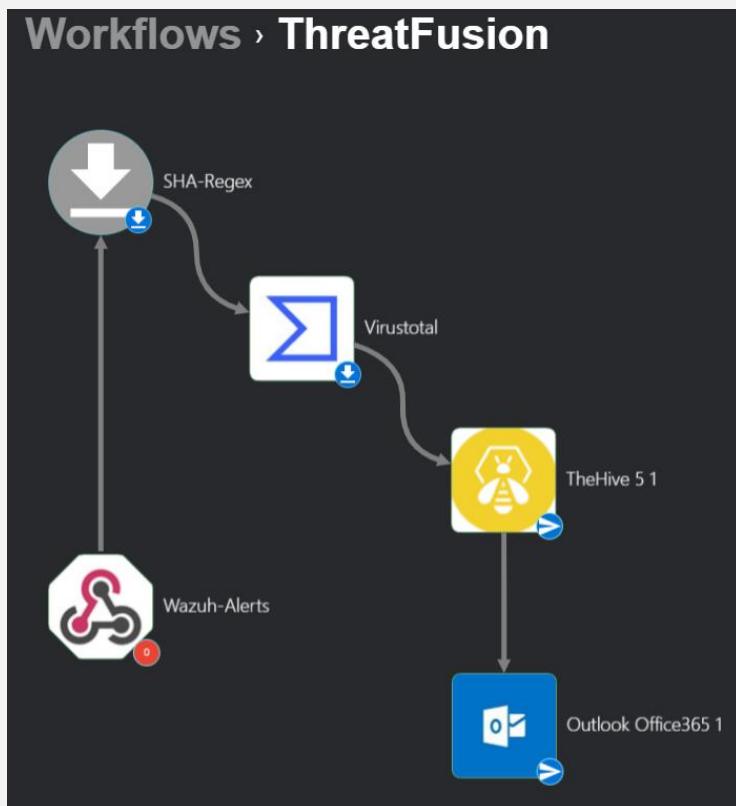
Security Alerts					
Time ↓	Technique(s)	Tactic(s)	Description	Level	Rule ID
> Apr 28, 2024 @ 11:41:46.857	T1003	Credential Access	Mimikatz Usage Detected	15	100002
Rows per page: 10 < 1 >					

data.win.eventdata.image	C:\Users\Vishwa Pancholi\Downloads\mimikatz_trunk\x64\Hackerspace.exe
data.win.eventdata.integrityLevel	High
data.win.eventdata.logonGuid	{1f7461ba-c7fd-662d-fa37-020000000000}
data.win.eventdata.logonId	0x237fa
🔍 📄 data.win.eventdata.originalFileName	mimikatz.exe

- It means we successfully generated an Alert in our dashboard.

6. Workflow Creation

- In the "Workflow Creation" section, I'll design a workflow utilizing the Shuffle tool, integrating Wazuh and TheHive. Within this workflow, I'll incorporate Virustotal to check the reputation score of detected events. Upon detection of Mimikatz usage by Wazuh, an alert will be dispatched to TheHive, facilitating seamless incident management. The SOC Analyst will then monitor the alert within TheHive, enabling timely investigation and response to security threats. Through this process, the SOC Analyst can efficiently leverage Wazuh, TheHive, and Virustotal to enhance threat detection and incident response capabilities.



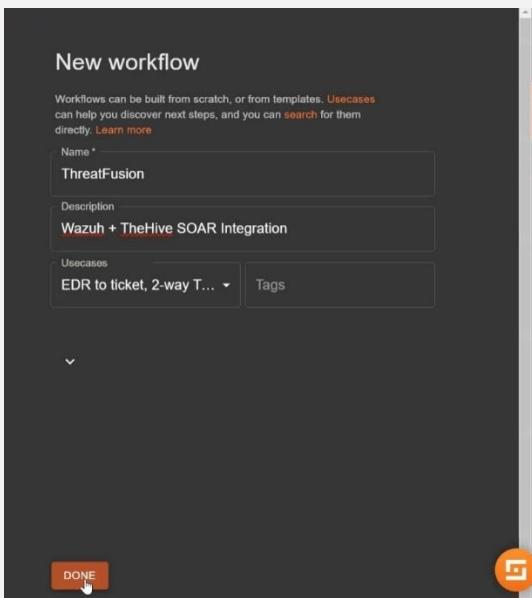
Workflow:

- 1) Mimikatz Alert Sent to Shuffle
- 2) Shuffle receives Mimikatz Alert(Extract SHA256 Hash from file)
- 3) Check Reputation Score with Virustotal
- 4) Send Details to TheHive to Create Alert
- 5) Send Email to SOC Analyst to Begin Investigation

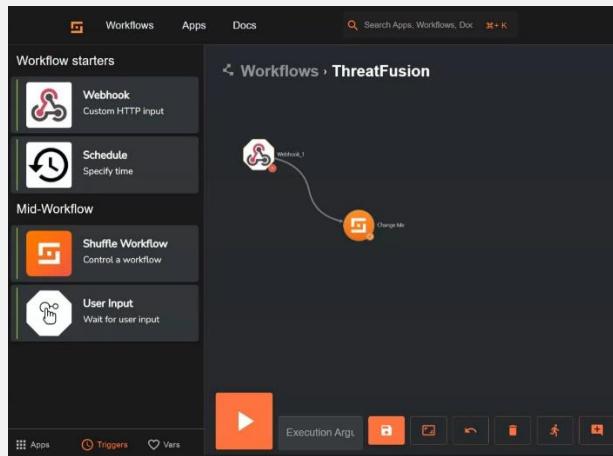
6.1 Handling Mimikatz Alerts: Shuffling Process

- To initiate the process of handling Mimikatz alerts through the Shuffle platform, we'll first create a Shuffle account. This involves signing up on the Shuffle platform's website and configuring account settings. Once the account is set up, we'll proceed to configure Wazuh to redirect alerts to Shuffle. This entails updating the Wazuh manager's configuration to include the Shuffle API endpoint and authentication credentials. Once configured, Wazuh alerts will be automatically forwarded to Shuffle, enabling centralized alert management and workflow automation.

- Step 1:



- We'll Create a workflow named ThreatFusion and add description.



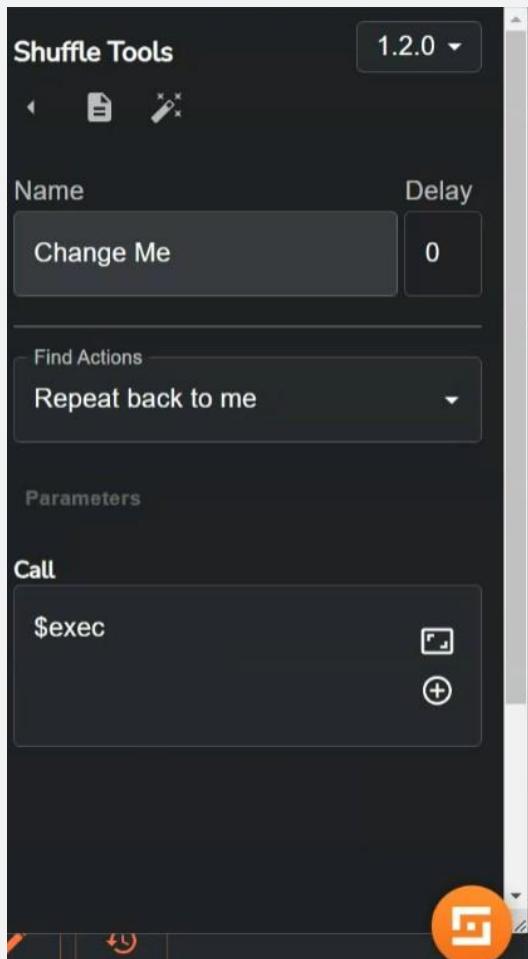
- We'll going to redirect the wazuh alert to shuffle for that you can see there's already change me icon is there we'll add new webhook for wazuh_alerts add that from the triggers

- Step 2:

The screenshot shows the ThreatFusion Webhook configuration page. The title is 'Webhook: uninitialized'. It includes sections for 'Name' (Wazuh-alerts), 'Find Associated App (optional)', 'Environment' (cloud), 'Parameters', and 'Webhook URI' (https://shuffler.io/api/v1/hooks/w). There are 'START' and 'STOP' buttons at the bottom, along with an 'Authentication headers' section and a 'Shuffle' icon.

- After adding the webhook we'll modify the name of it and after that save that

- Step 3:



- Now, we are going to modify the change me tool in that add Name, Actions and call which we are going to

- Step 4:

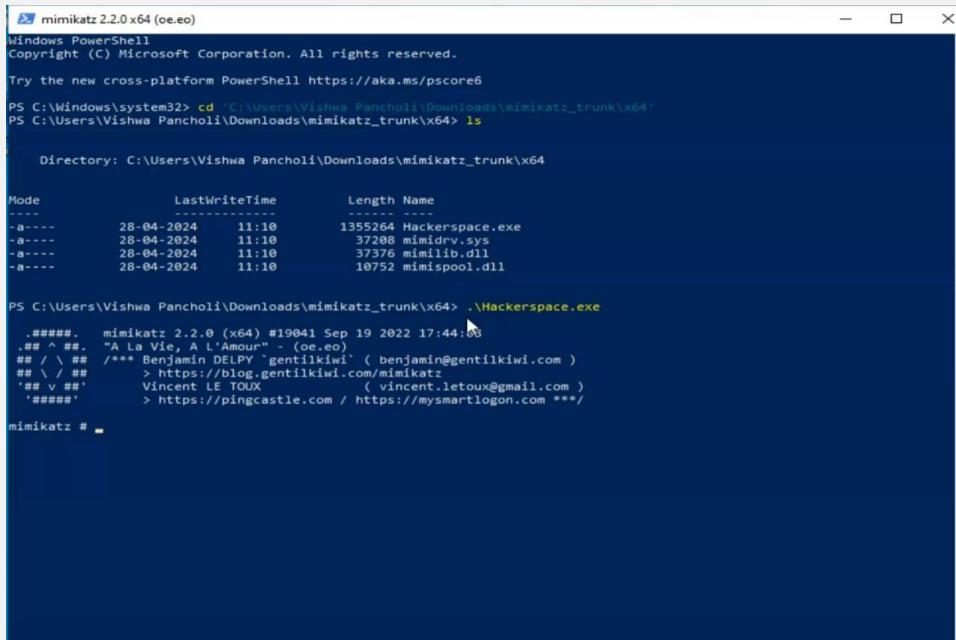
- Now, go to the wazuhmanager SSH and edit the ossec.conf file for that file path is /var/ossec/etc/ossec.conf

The terminal window shows the command: `root@wazuhmanager:~# nano /var/ossec/etc/ossec.conf`. The background of the terminal window is black, and the text is white.

- In that add the below Integration tag in it. Restart the wazuh-manager

```
<integration>
  <name>shuffle</name>
  <hook_url>https://shuffle.io/api/v1/hooks/webhook_2998f026-fae0-4496-b84c-56a2bef914bf </hook_url>
  <rule_id>100002</rule_id>
  <alert_format>json</alert_format>
</integration>
```

- Step 5:
- Now, go to the agents and run the mimikatz app which we previously named as a Hackerspace



The screenshot shows a Windows PowerShell window titled "mimikatz 2.2.0 x64 (oe.eo)". The command PS C:\Windows\system32> cd 'C:\Users\Vishwa Pancholi\Downloads\mimikatz_trunk\x64' is entered, followed by PS C:\Users\Vishwa Pancholi\Downloads\mimikatz_trunk\x64> ls. The output lists four files: Hackerspace.exe, mimidrv.sys, mimilib.dll, and mimispool.dll, all modified on 28-04-2024 at 11:10. The next command is PS C:\Users\Vishwa Pancholi\Downloads\mimikatz_trunk\x64> .\Hackerspace.exe, which starts the mimikatz application. The application's splash screen is visible, showing version 2.2.0 and copyright information.

- Step 6:
- Now, save and run the shuffle workflow and then you can see the results.

The screenshot shows the Wazuh Manager interface with two execution results:

- Execution Argument**: Status FINISHED, Source webhook, Started 30/04/2024, 20:25:00, Finished 30/04/2024, 20:25:01. The results show an alert titled "Mimikatz Usage Detected".
- Change Me**: Status SUCCESS, Source webhook, Started 30/04/2024, 20:25:01, Finished 30/04/2024, 20:25:01. The results show a Windows event log entry.

The "Execution Argument" result is expanded, showing the following JSON structure:

```

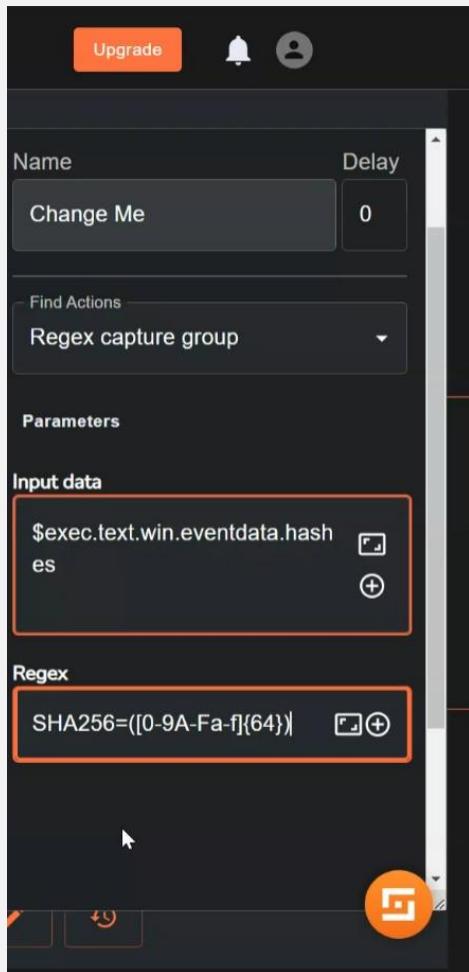
{
  "Results for Execution Argument": [
    {
      "severity": "3",
      "pretext": "WAZUH Alert",
      "title": "Mimikatz Usage Detected",
      "text": [
        {
          "win": [
            {
              "system": [
                {
                  "providerName": "Microsoft-Windows-Sysmon",
                  "providerGuid": "{5770385f-c22a-43e0-bf4c-06f5698ffbd9}",
                  "eventID": "1",
                  "version": "5",
                  "level": "4",
                  "task": "1",
                  "opcode": "0",
                  "keywords": "0x8000000000000000",
                  "systemTime": "2024-04-30T14:54:57.977841Z",
                  "eventRecordID": "5452",
                  "processID": "3688",
                  "threadID": "5056",
                  "channel": "Microsoft-Windows-Sysmon/Operational",
                  "computer": "DESKTOP-HTPA53F",
                  "severityValue": "INFORMATION",
                  "message": "A process with PID 3688 and Thread ID 5056 has initiated a Mimikatz session. This is a potential security risk as it may be used for credential theft or privilege escalation. It is recommended to investigate further and take appropriate action to mitigate this threat."
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}

```

- In the results you can see there's an Alert which we previously see in wazuh manager

6.2 Converting Alert to Hash

- In the process of converting alerts to a hash format, each incoming alert will undergo transformation into SHA256 hash. This conversion ensures a standardized representation of the alert data, facilitating efficient analysis and comparison. By utilizing the SHA256 hashing algorithm, we create a unique cryptographic hash value for each alert, enhancing data integrity and security within our workflow.
- For the conversion follow the below steps:
 - Step 1:
- Now, we'll change the name of Change me as a SHA_Regex and Action will be Regex capture group.
- Now, Regex is a sequence of characters that define a search pattern.



- In the input section we'll add \$execution argument -> select hash
- And here regex this value I got from the ChatGPT:

- Step 2:

- Now, save the workflow and run the execution you'll see the result your alert converted into the hash form.

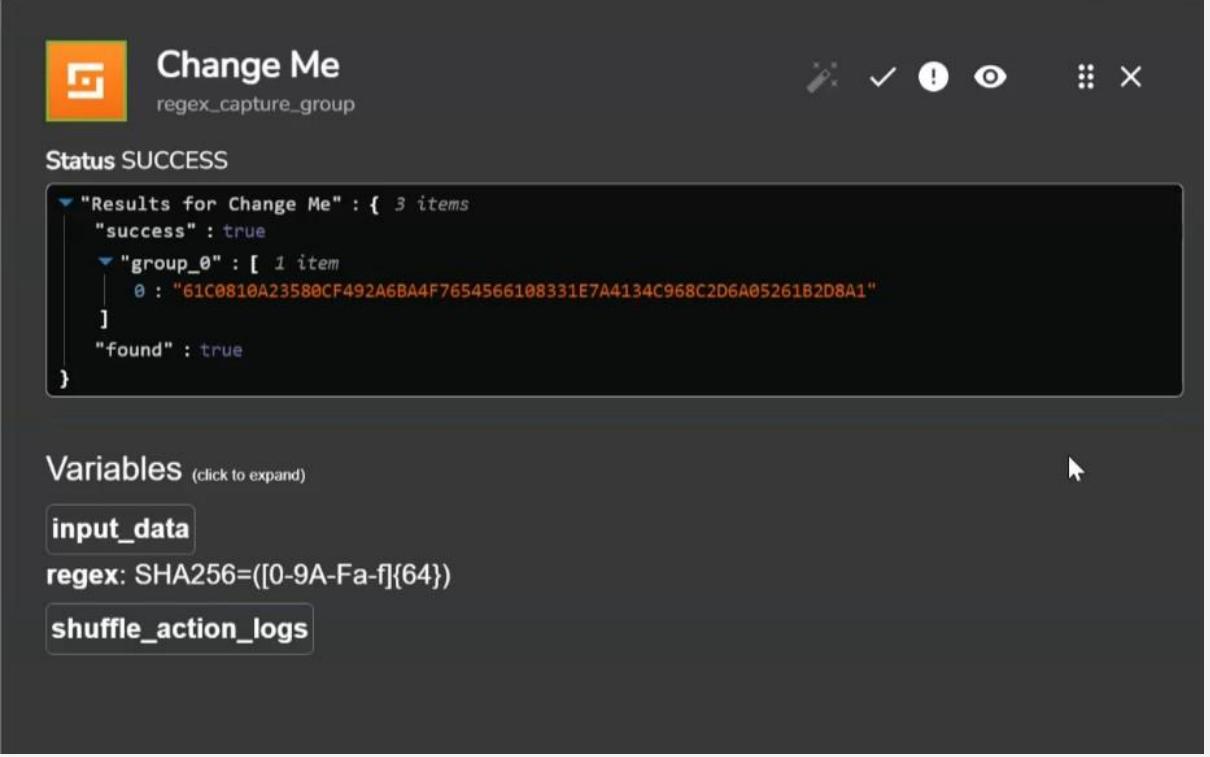
Execution Argument

Status SUCCESS

```

{
  "Results for Execution Argument": [
    {
      "severity": 3,
      "pretext": "WAZUH alert",
      "title": "Mimikatz Usage Detected",
      "text": [
        {
          "win": [
            {
              "system": [
                {
                  "providerName": "Microsoft-Windows-Sysmon",
                  "providerGuid": "{5770385f-c22a-43e0-bf4c-06f5698ffbd9}",
                  "eventID": "1",
                  "version": "5",
                  "level": "4",
                  "task": "1",
                  "opcode": "0",
                  "keywords": "0x8000000000000000",
                  "systemTime": "2024-04-30T14:54:57.9778411Z",
                  "eventRecordID": "5452",
                  "processID": "3688",
                  "threadID": "5056",
                  "channel": "Microsoft-Windows-Sysmon/Operational",
                  "computer": "DESKTOP-HPTA53F",
                  "severityValue": "INFORMATION",
                  "message": ""
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
  
```

- Step 3:
 - Here you can see the hash formed body of your alert.



The screenshot shows a user interface for a search or analysis tool. At the top, there is a logo with a stylized orange 'G' and the text "Change Me" followed by "regex_capture_group". To the right are several icons: a pencil, a checkmark, an exclamation mark, an eye, three dots, and an 'X'. Below this, the status is displayed as "Status SUCCESS". The main content area contains a JSON-like data structure:

```

{
  "Results for Change Me": {
    "success": true,
    "group_0": [
      {
        "0": "61C0810A23580CF492A6BA4F7654566108331E7A4134C968C2D6A05261B2D8A1"
      }
    ],
    "found": true
  }
}
  
```

Below this, under the heading "Variables (click to expand)", are three entries:

- input_data**
- regex**: SHA256=([0-9A-Fa-f]{64})
- shuffle_action_logs**

6.3 Verifying Reputation Score via Virustotal

- After converting the alert into a hash format, we will proceed to verify its reputation score using Virustotal. Virustotal is a comprehensive online service that aggregates data from various antivirus engines, website scanners, and file analysis tools to assess the reputation of files and URLs. It provides insights into the trustworthiness and potential threats associated with a given hash or URL. By leveraging Virustotal's extensive database and analysis capabilities, we can augment our threat intelligence efforts and make informed decisions regarding the severity of detected alerts.

- Step 1:

- Firstly, create an account on Virustotal and copy the API key.

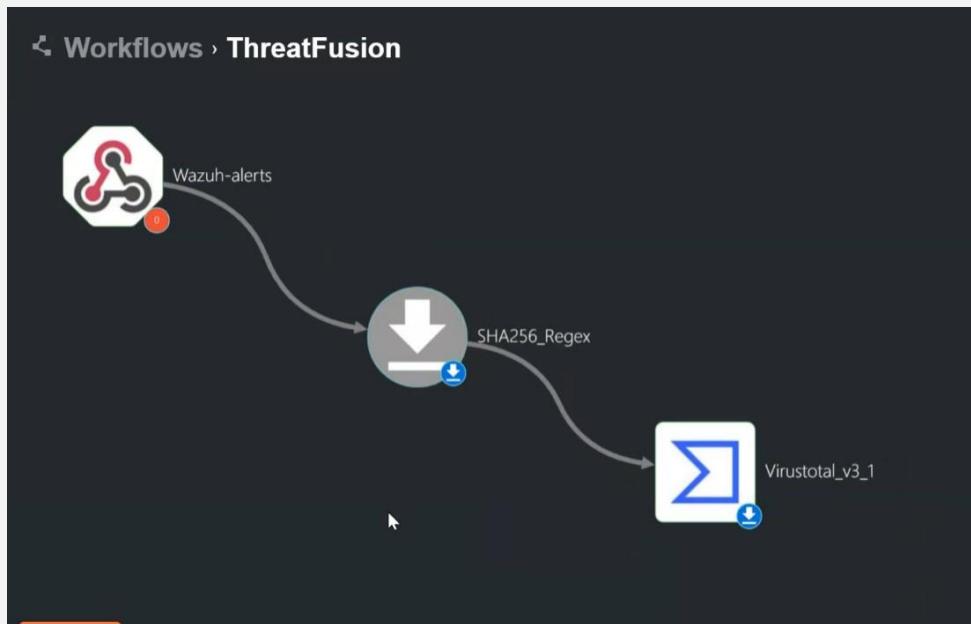
The screenshot shows the VirusTotal API Key page. At the top, it displays the API key itself, which is heavily redacted. Below this, the section 'API QUOTA ALLOWANCES FOR YOUR USER' provides details about the user's account level (Limited, standard free public API), usage restrictions (Must not be used in business workflows, commercial products or services), and request rate limits (4 lookups/min, 500 lookups/day, 15.5 K lookups/month). It also includes links to upgrade to premium, use in browser, discover feeds, and other services. The bottom section, 'CONSUMPTION LAST 30 DAYS', shows a graph of consumption over time.

- Now, in the previous flow we get the hash value copy paste that hash value in the Virustotal and at there you can check the reputation score of that alert.

The screenshot shows the VirusTotal analysis page for the file 'mimikatz.exe' with the hash '61c0810a23580cf492a6ba4f7f654566108331e7a4134c968c2d6a05261b2d8a1'. The main interface displays a 'Community Score' of 64/72. The file details include its name, size (1.29 MB), last modification date (53 minutes ago), and file type (EXE). The 'DETECTION' tab is selected, showing numerous detection rules from various sources, such as 'HCKL_Mimikatz_SkeletonKey_in_memory_Aug20_1' and 'HCKL_mimikatz_icon'. The 'CROWDSOURCED YARA RULES' and 'CROWDSOURCED SIGMA RULES' sections also provide additional context on the file's behavior.

- Step 2:

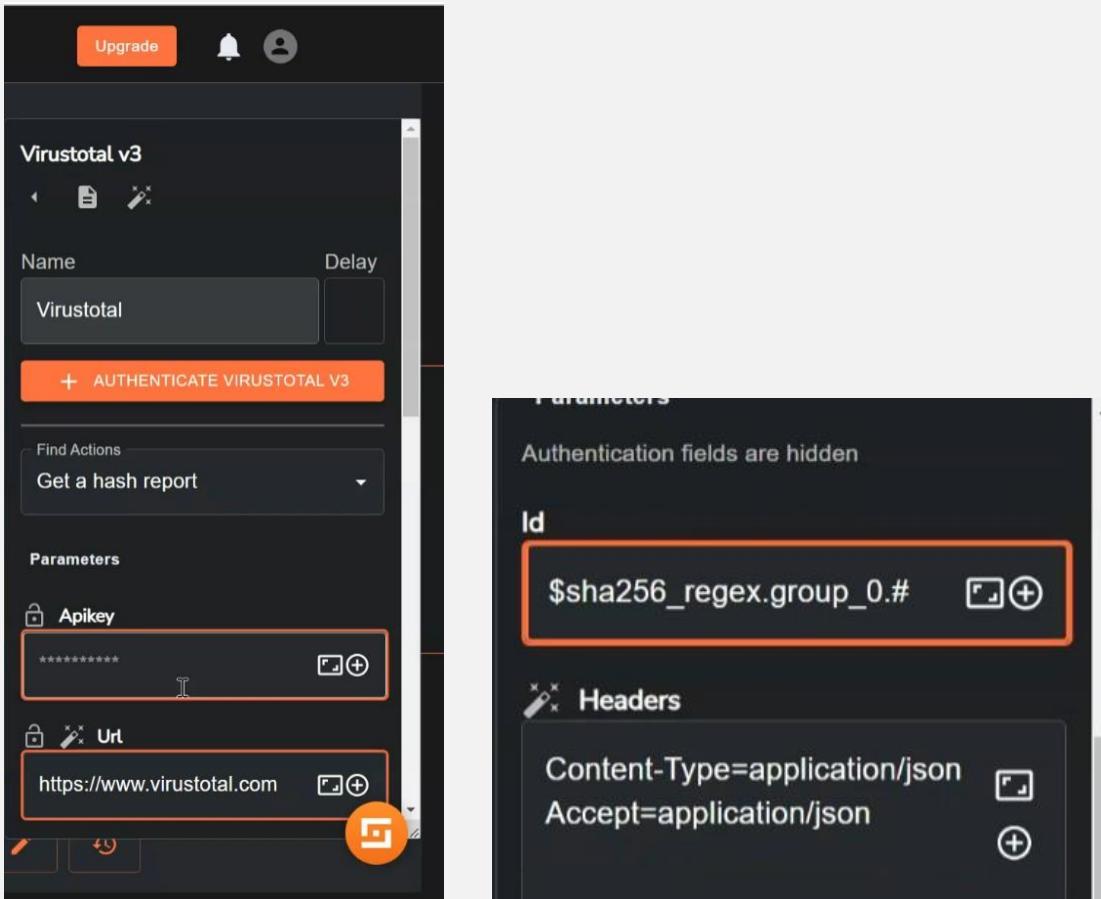
- Now, in the shuffle activate the Virustotal app and add it in the workflow.



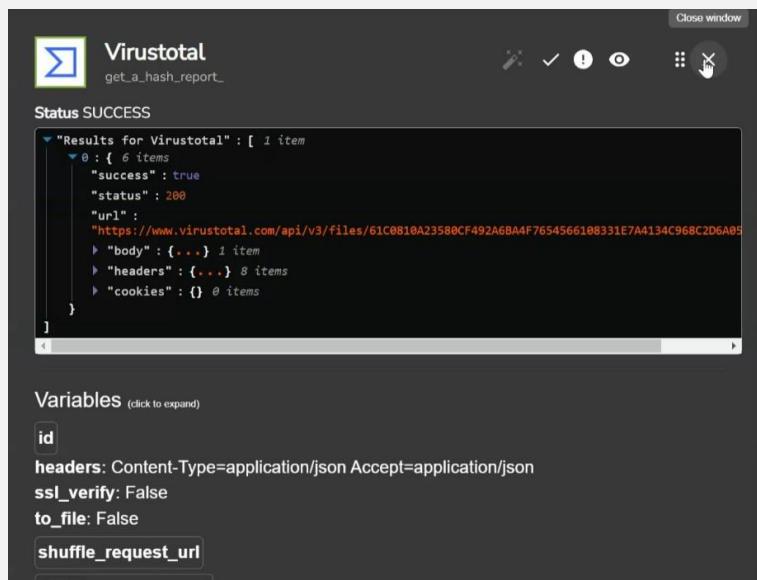
- Now, Authenticate the Virustotal App using the API key which you got in Virustotal app.

The image contains two side-by-side screenshots. The left screenshot is a 'Authentication for Virustotal v3' dialog. It includes fields for 'Name - what is this used for?' (set to 'Auth for Virustotal_v3'), 'apikey' (with a redacted value), and 'url' (set to 'https://www.virustotal.com'). There are 'CANCEL' and 'SUBMIT' buttons at the bottom. The right screenshot shows the 'Virustotal V3' dashboard. It features a user profile for 'Emi Martine...' with a picture of a dog, and a 'Profile' section. Below the profile is a link labeled 'API key'.

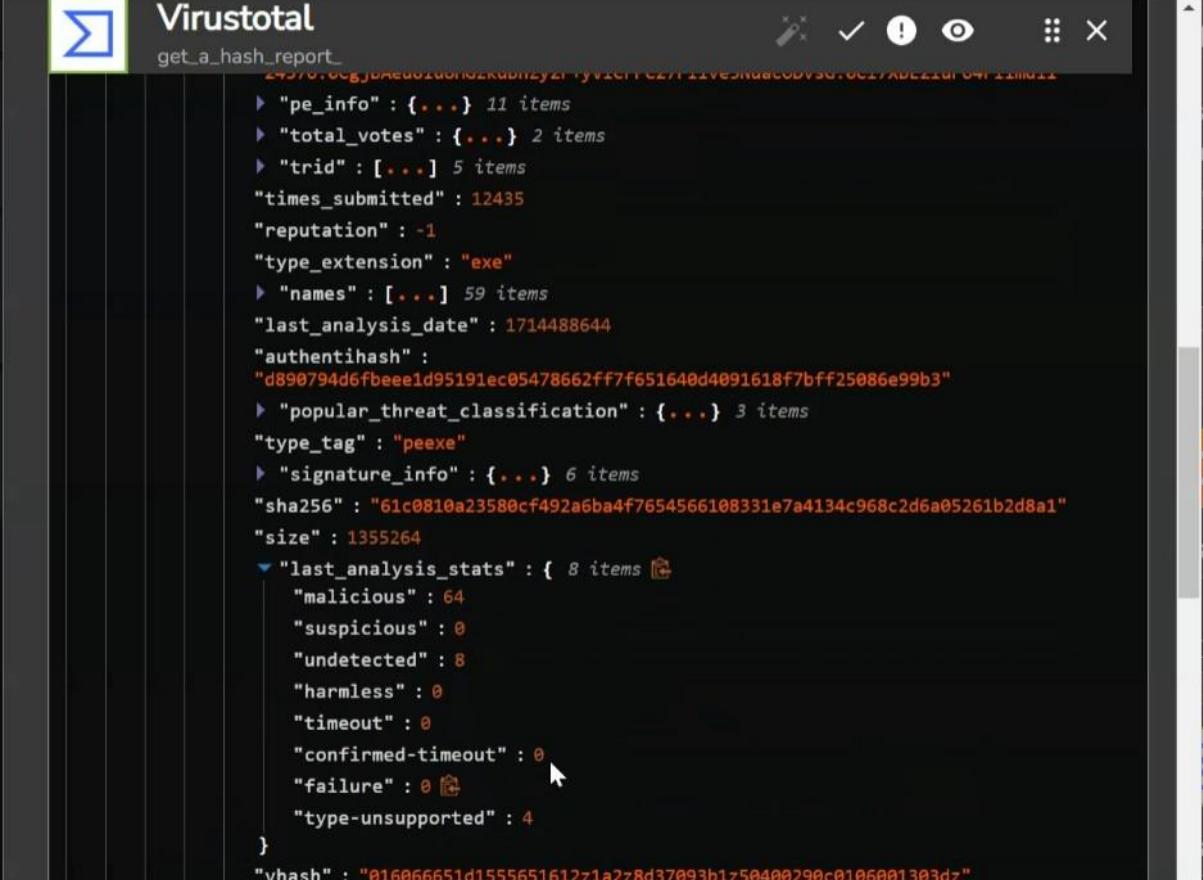
- Step 3:
 - Now, in the ID section you'll add the execution sha_regex.



- Step 4:
 - Now, save the workflow and run execution and here you got the URL which redirect you to the Virustotal.



- And also, in the execution result you'll also see the malicious rate under body -> last_analysis_stats



```

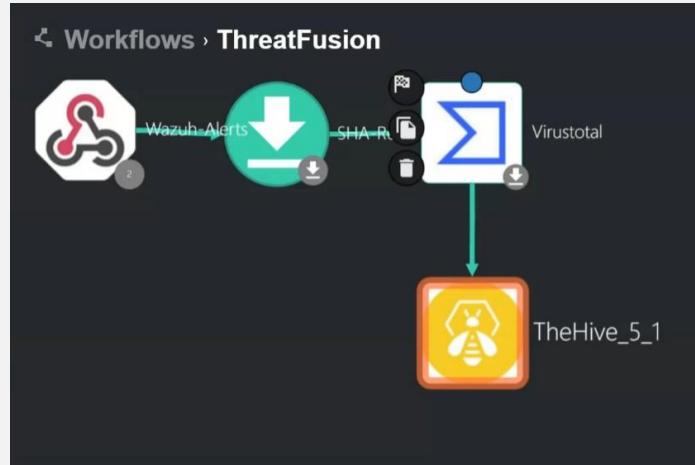
{
  "pe_info": {...} 11 items
  "total_votes": {...} 2 items
  "trid": [...] 5 items
  "times_submitted": 12435
  "reputation": -1
  "type_extension": "exe"
  "names": [...] 59 items
  "last_analysis_date": 1714488644
  "authentihash": "d890794d6fbee1d95191ec05478662ff7f651640d4091618f7bff25086e99b3"
  "popular_threat_classification": {...} 3 items
  "type_tag": "peexe"
  "signature_info": {...} 6 items
  "sha256": "61c0810a23580cf492a6ba4f7654566108331e7a4134c968c2d6a05261b2d8a1"
  "size": 1355264
  "last_analysis_stats": {...} 8 items
    "malicious": 64
    "suspicious": 0
    "undetected": 8
    "harmless": 0
    "timeout": 0
    "confirmed-timeout": 0
    "failure": 0
    "type-unsupported": 4
}
  "vhash": "016066651d1555651612z1a2z8d37093b1z50400290c0106001303dz"
}

```

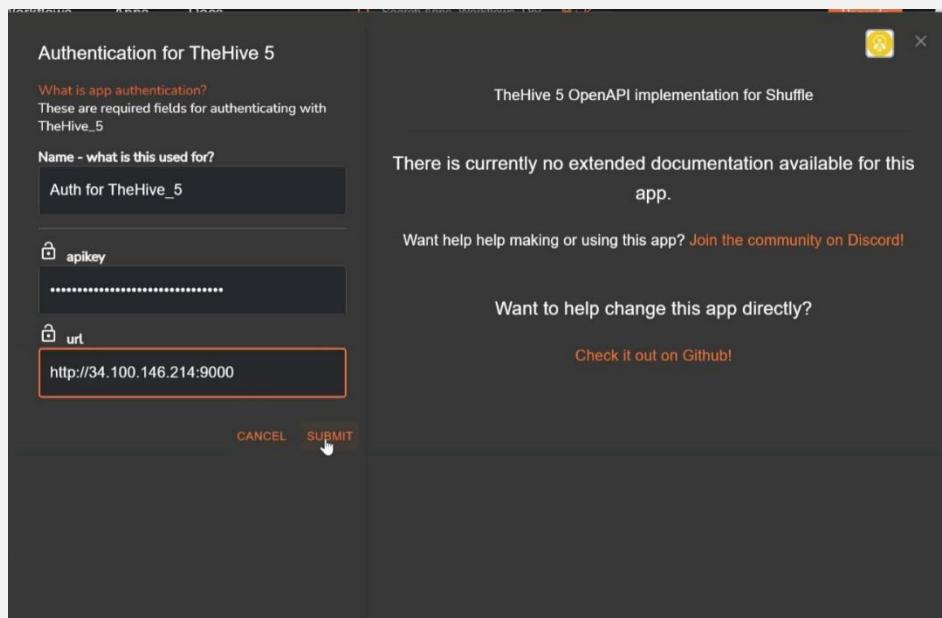
6.4 Creating Alerts in TheHive

- Upon successful redirection of Wazuh alerts, they will be created as alerts within TheHive user interface. TheHive serves as a centralized platform for incident management and response, allowing security analysts to view, triage, and investigate alerts efficiently. By integrating Wazuh with TheHive, we streamline the process of alert handling, enabling seamless collaboration and workflow automation for effective incident response.

- Step 1:
 - Add the TheHive app in the workflow.



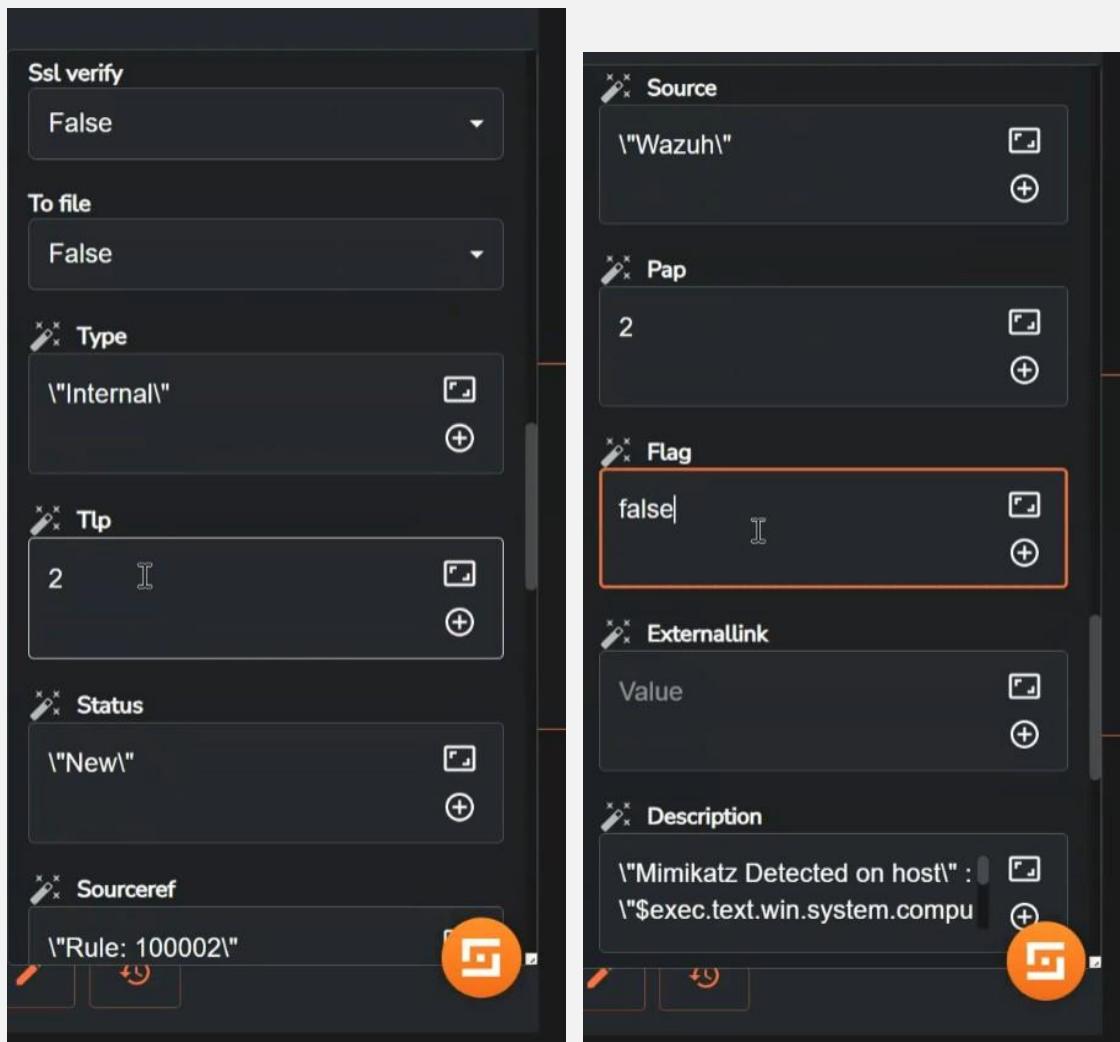
- Now, authenticate the app using SOAR user API and in URL section add TheHive Public IP Address and Port number 9000.



- Step 2:
 - Now, modify the configurations as per you need here I'm adding Tag, Summary, Severity, Title,Summary, Status and description.

The screenshot shows a configuration interface with several sections:

- Show Body**: A checkbox labeled "Show Body" with an orange checkmark.
- Title**: A field containing the value "\\$exec.title\".
- Tags**: A field containing the value ["T10003"].
- Summary**: A field containing the value "Mimikatz activity detected on host".
- Severity**: A field containing the value 2.
- Headers**: A field containing the values Content-Type=application/json and Accept=application/json.



- Step 3:

- Here, you can see the alert in the Vishwa User.

Alerts		Enter a case number		CREATE CASE	ENGLISH (UK)	VISHWA	Clear filters
<input type="button" value="default"/>	<input type="button" value="Quick Filters"/>	<input type="button" value="Export list"/>					
stage: any(New) X							

STATUS	SEVERITY	TITLE	# CASE	TYPE	SOURCE	REFERENCE	DETAILS	ASSIGNEE	DATES	O.	C.	U.
New	Info	Mimikatz Usage Detected	T10003	Internal	Observables				03/05/2024 05:30			

Details for T10003:

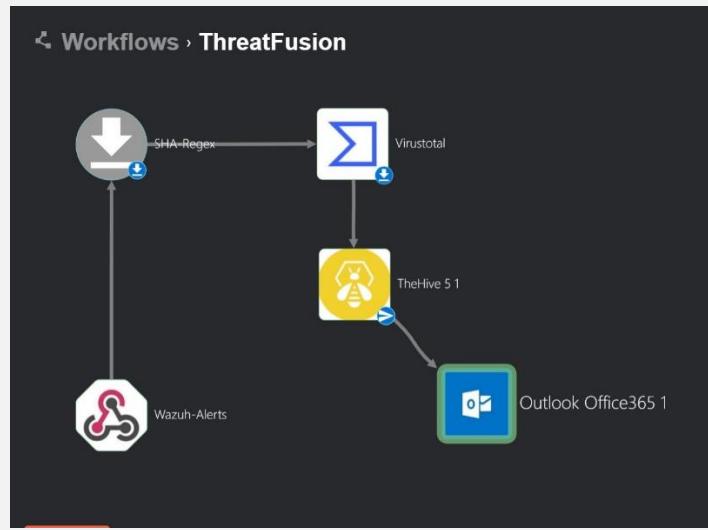
- Source: Wazuh
- Reference: Rule: 100002
- None

The screenshot shows a detailed view of an alert in TheHive. The alert title is "Mimikatz Usage Detected". Key details include:

- General Tab:** Shows the alert ID (~42102856), created by SOAR at 03/05/2024 08:56. It has a severity of MEDIUM, TLP:AMBER, and PAP:AMBER.
- Tags:** T10003
- Description:** "Mimikatz Detected on host" : "DESKTOP-HPTA53F" "from user" : "DESKTOP-HPTA53F\Vishwa Pancholi"
- Summary:** "Mimikatz activity detected on host" : "DESKTOP-HPTA53F" "and the process ID is" : "3284" "and the Command is" : "'C:\Users\Vishwa Pancholi\Downloads\mimikatz_trunk\x64\Hackerspace.exe'"
- Assignee:** Unassigned
- Source:** Wazuh
- Reference:** Rule: 100002
- Type:** Internal
- Occurred date:** 03/05/2024 05:30
- Status:** New

6.5 Email Notification for SOC Analysts to Initiate Investigation

- Upon detection of an alert within TheHive, an email notification will be automatically generated and sent to the SOC Analysts, providing them with pertinent information regarding the alert. This email notification serves as a prompt for SOC Analysts to initiate investigation and response activities. By receiving alerts in their email inbox, SOC Analysts can promptly monitor and assess the alert's severity, enabling timely and effective incident response.
- Step 1:
 - Add the Outlook office 365 App in the workflow



- Step 2:

- Now, authenticate using One-Click Login and enter your email and code it will authenticate automatically.

Authenticate outlook office365

Oauth2 requires a client ID and secret to authenticate, defined in the remote system. Your redirect URL is https://shuffler.io/set_authentication - [Learn more about OAuth2 with Shuffle](#)

One-click Login OR
url: https://graph.microsoft.com
Client ID
Client Secret
Scopes (access rights)
AUTHENTICATE

Outlook Office365

Microsoft Graph

To connect with and receive emails from Office365, you'll need Oauth2. This app can get and update emails as you wish. Mail functions from the Microsoft Graph API. This process will help you want to connect to your organisation locally or with our cloud. To security constraints of Oauth2 authentication, this is required.

Authentication

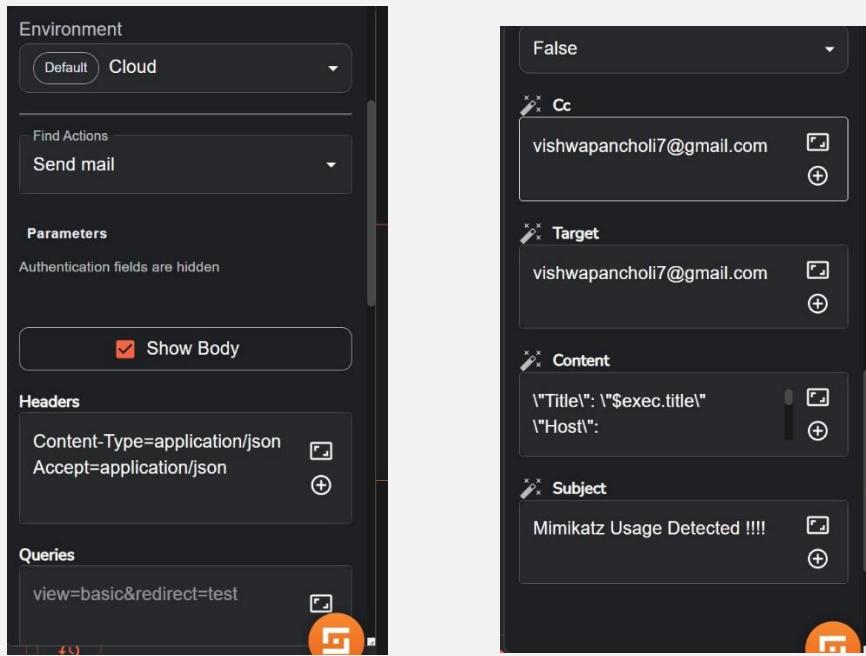
To authenticate this app, you'll need an app registered in your organization. You should use what's called "delegated permissions", NOT "Application permissions". More about this further down.

Required:

- tenant_id

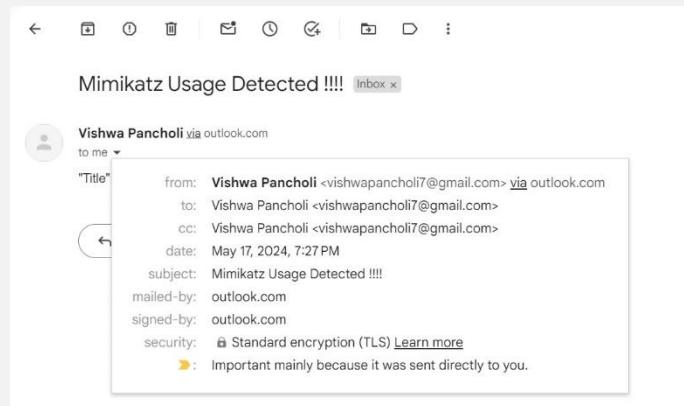
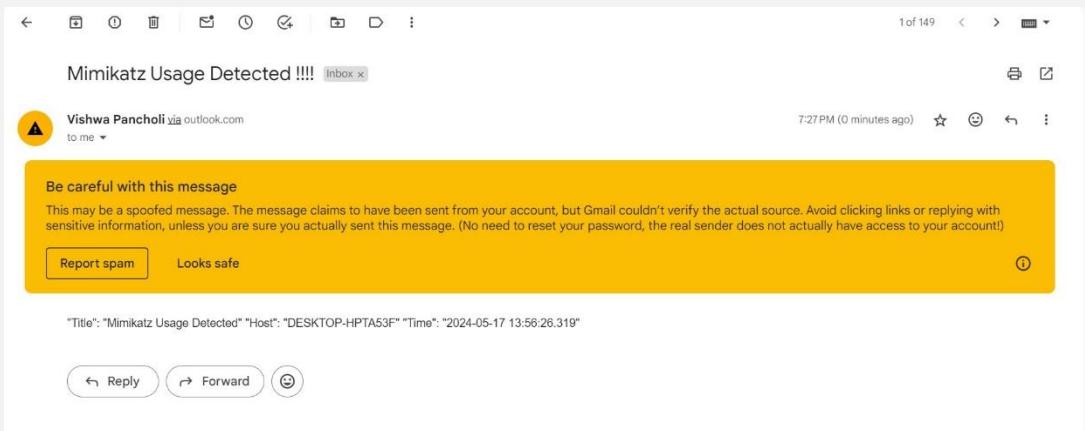
- Step 3:

- Now, in the action add send email option. Add Subject and Content; and also add the email of SOC Analyst in recipient and cc.



- Step 4:

- Now, Save the workflow and run the execution mean while go to your windows agent add rerun the hackerspace.exe file and You'll get the email in your inbox named Mimikatz Usage Detected !!!



Conclusion

In conclusion, the integration of Wazuh and TheHive, combined with workflow automation using tools like Shuffle and Virustotal, enhances our ability to detect, triage, and respond to security threats effectively. By centralizing alert management, streamlining incident response workflows, and providing timely notifications to SOC Analysts, we empower organizations to bolster their cybersecurity posture and mitigate risks proactively. This project demonstrates the importance of leveraging comprehensive security solutions and automation tools to safeguard digital assets and ensure the resilience of modern IT environments.

REFERENCES

- Wazuh:
 - <https://github.com/MyDFIR/SOC-Automation-Project/blob/main/Wazuh-Install-Instructions>
- TheHive:
 - <https://github.com/ls111-cybersec/thehive-cortex-misp-docker-compose-lab11update/blob/main/docker-compose.yml>
- Sysmon:
 - <https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon>
<https://github.com/olafhartong/sysmon-modular/blob/master/sysmonconfig.xml>
- Mimikatz:
 - <https://github.com/gentilkiwi/mimikatz/releases>
- <https://www.youtube.com/@MyDFIR>