SOC Platform Deployment Guide

Security Operations Center Technologies

TOOLS USED:

VIRTUAL BOX	VirtualBox is a general-purpose full virtualizer for x86 hardware, targeted at server, desktop and embedded use.
UBUNTU	Ubuntu aims to provide a user-friendly, stable, and secure operating system for a wide range of computing needs, from personal desktops to enterprise servers and cloud environments.
WAZUH	Wazuh help organizations detect and respond to threats, ensure compliance, and perform security analytics.
OPEN VAS	OpenVAS (Open Vulnerability Assessment Scanner) is an open-source vulnerability scanner designed to help organizations identify security vulnerabilities in their networks, systems, and applications.
THE HIVE	TheHive is an open-source Security Incident Response Platform designed to help security teams manage and respond to security incidents more effectively. It's a powerful tool for handling threats and coordinating incident response activities.
CORTEX	Cortex is an open-source and free software developed by TheHive Project to analyze observables (such as IP addresses, email addresses, URLs, domain names, files, and hashes) and provide actionable intelligence.
MISP	MISP (Malware Information Sharing Platform) is an open-source threat intelligence platform designed to help organizations share, store, and correlate information about cyber threats. Here are some key features of MISP.
KIBANA	Kibana is a data visualization and exploration tool used for log and time-series analytics, application monitoring, and operational intelligence use cases.

Installation Steps

VIRTUAL BOX

Prerequisite:

Supported Host Operating Systems:

Windows hosts (64-bit)	**Windows 8.1**
	Windows 10
	Windows 11 21H2
	Windows Server 2012
	Windows Server 2012 R2
	Windows Server 2016
	Windows Server 2019
	Windows Server 2022

Intel hardware is required.

Host CPU Requirements:

SSE2 (Streaming SIMD Extensions 2) support is required for host CPUs.

Installing Oracle VM VirtualBox and Extension Packs:

Oracle VM VirtualBox comes in many different packages, and installation depends on your host OS.

Base package: The base package consists of all open source components and is licensed under the GNU General Public License V3.

Extension packs: Additional extension packs can be downloaded which extend the functionality of the Oracle VM VirtualBox base package. Currently, Oracle provides a single extension pack, available from: http://www.virtualbox.org.

Performing the Installation:

The Oracle VM VirtualBox installation can be started in either of the following ways:

• **By double-clicking on the executable file.**

UBUNTU

Installation in VirtualBox:

Performing installation in VirtualBox:

1. Download ISO for Ubuntu.
2. Open VirtualBox and click “New” button.
3. Give a **Name** to your **Virtual Machine** and select the **Location** for it to install.
4. Assign **RAM Size** to your Virtual Machine.
5. Create a **Virtual Hard Disk** for the machine to store files.
6. Select the type of Hard disk. Using **VDI** type is recommended.
7. Either of the **Physical Storage** types can be selected. Using a **Dynamically Allocated Disk** is by default recommended.
8. Select **Disk Size** and provide the **Destination Folder** to install.
9. After the Disk creation is done, boot the **Virtual Machine** and begin installing **Ubuntu**.
10. If the installation disk is not automatically detected. Browse the file location and select the **ISO file for Ubuntu**.
11. Proceed with the installation file and wait for further options.
12. Click on the **Install Ubuntu** option, this might look different for other Ubuntu versions.
13. Select **Keyboard Layout**, if the defaults are compatible, just click on the **Continue** button and proceed.
14. Select **Installation Type**. By default, it is set to **Normal Installation**, which is recommended, but it can also be changed to **Minimal Installation** if there is no need for all Ubuntu features.
15. Click on the **Install Now** button and carry on with the installation. Do not get worried about the **Erase disk** option, it will only be effective inside the virtual machine, and other system files outside the VirtualBox remain intact.
16. Click on the **Continue** button, and proceed with writing changes on the disk.
17. Select your Location to set the **Time Zone**.
18. Choose a **Name** for your computer and set a **Password** to secure login info.
19. Once the installation process is over, reboot your Virtual Machine.
20. You’re finished with the installation process. Now you can use Ubuntu along with Windows, without creating a dual boot.

WAZUH

Wazuh provides a pre-built virtual machine image in Open Virtual Appliance (OVA) format. This can be directly imported to VirtualBox or other OVA compatible virtualization systems.

Download the virtual appliance (OVA), which contains the following components:

- Wazuh manager 4.9.0
- Wazuh indexer 4.9.0
- Filebeat-OSS 7.10.2
- Wazuh dashboard 4.9.0

Hardware requirements:

The following requirements have to be in place before the Wazuh VM can be imported into a host operating system:

- The host operating system has to be a 64-bit system.
- Hardware virtualization has to be enabled on the firmware of the host.
- A virtualization platform, such as VirtualBox, should be installed on the host system.

Installation in virtual machine:

1. Import the OVA to the virtualization platform.
2. If you're using VirtualBox, set the VMSVGA graphic controller. Setting another graphic controller freezes the VM window.
    - Select the imported VM.
    - Click **Settings** > **Display**
    - In **Graphic controller**, select the VMSVGA option.
3. Start the machine.
4. Access the virtual machine using the following user and password. You can use the virtualization platform or access it via SSH.

    user: wazuh-user

    password: wazuh

    SSH root user login has been deactivated; nevertheless, the wazuh-user retains sudo privileges. Root privilege escalation can be achieved by executing the following command:

sudo -i

Access the Wazuh dashboard:

1. Shortly after starting the VM, the Wazuh dashboard can be accessed from the web interface by using the following credentials:

   URL: https://<wazuh_server_ip>
   user: admin
   password: admin

2. You can find <wazuh_server_ip> by typing the following command in the VM:

   ip a

Configuration files:

All components included in this virtual image are configured to work out-of-the-box, without the need to modify any settings. However, all components can be fully customized. These are the configuration files locations:

1. Wazuh manager: /var/ossec/etc/ossec.conf
2. Wazuh indexer: /etc/wazuh-indexer/opensearch.yml
3. Filebeat-OSS: /etc/filebeat/filebeat.yml
4. Wazuh dashboard:
   - /etc/wazuh-dashboard/opensearch_dashboards.yml
   - /usr/share/wazuh-dashboard/data/wazuh/config/wazuh.yml

VIRTUALBOX TIME CONFIGURATION:

Once the virtual machine is imported it may run into issues caused by time skew when VirtualBox synchronizes the time of the guest machine. To avoid this situation, enable the Hardware Clock in UTC Time option in the System tab of the virtual machine configuration.

Once the virtual machine is imported and running, the next step is to deploy the Wazuh agents on the systems to be monitored.

OPEN VAS

PREREQUISITES:

Before proceeding with the installation, ensure that the following program is already installed on your system:

1. Ubuntu 20.04/22.04

UPDATE SYSTEM:

sudo apt update && sudo apt upgrade -y

INSTALL OPENVAS:

sudo apt install -y gvm

SETUP OPEN VAS:

1. Initialize OpenVAS (this may take some time):

sudo gvm-setup

START OPEN VAS:

sudo gvm-start

GET ADMIN CREDENTIALS:

1. To retrieve the default admin password:

sudo gvm-admin --get-users

ACCESS WEB INTERFACE:

1.Open your browser and go to:

https://127.0.0.1:9392

2.Use the credentials obtained earlier to log in.

Running and Using OpenVAS:

Once installed and running, follow these steps to perform a vulnerability scan:

1. **Log in to the Web UI** at https://127.0.0.1:9392.
2. **Create a Target**: Go to *Configuration* → *Targets* and define the IP range you want to scan.
3. **Start a Scan**: Navigate to *Scans* → *Tasks* and create a new task.
4. **Analyze Results**: After the scan completes, view the report under *Scans* → *Reports*.

THE HIVE

PREREQUISITES:

Before proceeding with the installation, ensure that the following programs are already installed on your system:

1. Open a terminal window.
2. Run the following command to install the necessary dependencies:

apt install wget gnupg apt-transport-https git ca-certificates ca-certificates-java curl software-properties-common python3-pip lsb-release

Ensure that all dependencies are successfully installed before proceeding with the TheHive installation process.

Java Virtual Machine:

1. Open a terminal window.
2. Execute the following commands:

wget -qO- https://apt.corretto.aws/corretto.key | sudo gpg --dearmor -o /usr/share/keyrings/corretto.gpg

echo "deb [signed-by=/usr/share/keyrings/corretto.gpg] https://apt.corretto.aws stable main" | sudo tee -a /etc/apt/sources.list.d/corretto.sources.list

sudo apt update

sudo apt install java-common java-11-amazon-corretto-jdk

echo JAVA_HOME="/usr/lib/jvm/java-11-amazon-corretto" | sudo tee -a /etc/environment

export JAVA_HOME="/usr/lib/jvm/java-11-amazon-corretto"

3. Verify the installation by running:

java -version

4. You should see output similar to the following:

openjdk version "11.0.12" 2022-07-19

OpenJDK Runtime Environment Corretto-11.0.12.7.1 (build 11.0.12+7-LTS)

OpenJDK 64-Bit Server VM Corretto-11.0.12.7.1 (build 11.0.12+7-LTS, mixed mode)

Apache Cassandra Installation:

Apache Cassandra is a highly scalable and robust database system. TheHive is fully compatible with Apache Cassandra's latest stable release version 4.0.x.

1. Add Apache Cassandra repository references:
- Download Apache Cassandra repository keys using the following command:

wget -qO - https://downloads.apache.org/cassandra/KEYS | sudo gpg --dearmor -o /usr/share/keyrings/cassandra-archive.gpg

- Add the repository to your system by appending the following line to the /etc/apt/sources.list.d/cassandra.sources.list file. This file may not exist, and you may need to create it.

    echo "deb [signed-by=/usr/share/keyrings/cassandra-archive.gpg] https://debian.cassandra.apache.org 40x main" | sudo tee -a

2. Install the package:

- Once the repository references are added, update your package index and install Cassandra using the following commands:

    sudo apt update

    sudo apt install cassandra

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

Configuration:

You can configure Cassandra by modifying settings within the */etc/cassandra/cassandra.yaml* file.

1.Locate the Cassandra Configuration File:

- Navigate to the directory containing the Cassandra configuration file */etc/cassandra/*.

2.Edit the *cassandra.yaml* File:

- Open the *cassandra.yaml* file in a text editor with appropriate permissions.

3.Configure Cluster Name:

- Set the *cluster_name* parameter to the desired name. This name helps identify the Cassandra cluster.

4.Configure Listen Address:

- Set the *listen_address* parameter to the IP address of the node within the cluster. This address is used by other nodes within the cluster to communicate.

5.Configure RPC Address:

- Set the *rpc_address* parameter to the IP address of the node to enable clients to connect to the Cassandra cluster.

6.Configure Seed Provider:

- Ensure the *seed_provider* section is properly configured. The seeds parameter should contain the IP address(es) of the seed node(s) in the cluster.

7.Configure Directories:

- Set the directories for data storage, commit logs, saved caches, and hints as per your requirements. Ensure that the specified directories exist and have appropriate permissions.

8.Save the Changes:

- After making the necessary configurations, save the changes to the *cassandra.yaml* file.

/etc/cassandra/cassandra.yaml

```
# content from /etc/cassandra/cassandra.yaml
[..]
cluster_name: 'thp'
listen_address: 'xx.xx.xx.xx' # address for nodes
rpc_address: 'xx.xx.xx.xx' # address for clients
seed_provider:
    - class_name: org.apache.cassandra.locator.SimpleSeedProvider
    parameters:
        # Ex: "<ip1>,<ip2>,<ip3>"
        - seeds: 'xx.xx.xx.xx' # self for the first node
data_file_directories:
- '/var/lib/cassandra/data'
commitlog_directory: '/var/lib/cassandra/commitlog'
saved_caches_directory: '/var/lib/cassandra/saved_caches'
hints_directory:
- '/var/lib/cassandra/hints'
[..]
```

Start the service:

1.Start the Service

- Execute the following command to start the Cassandra service:

sudo systemctl start cassandra

2. Ensure Service Restarts After Reboot:

- Enable the Cassandra service to restart automatically after a system reboot:

sudo systemctl enable cassandra

3.(Optional) Remove Existing Data Before Starting:

- If the Cassandra service was started automatically before configuring it, it's recommended to stop it, remove existing data, and restart it once the configuration is updated. Execute the following commands:

sudo systemctl stop cassandra

*sudo rm -rf /var/lib/cassandra/**

NOTE:

Cassandra defaults to listening on port 7000/tcp for inter-node communication and port 9042/tcp for client communication.

Elasticsearch:

Elasticsearch is a robust data indexing and search engine. It is used by TheHive to manage data indices efficiently.

NOTE:

From Version 5.3, TheHive supports Elasticsearch 8.0 and 7.x. Previous TheHive versions only support Elasticsearch 7.x.

Starting from TheHive 5.3, for advanced use-cases, OpenSearch is also supported.

Installation:

1. Add Elasticsearch repository references:

- To add Elasticsearch repository keys, execute the following command:

  wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo gpg --dearmor -o /usr/share/keyrings/elasticsearch-keyring.gpg

  sudo apt-get install apt-transport-https

- Add the repository to your system by appending the following line to the /etc/apt/sources.list.d/elastic-7.x.list file. This file may not exist, and you may need to create it

  echo "deb [signed-by=/usr/share/keyrings/elasticsearch-keyring.gpg] https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee /etc

2. Install the package:

- Once the repository references are added, update your package index and install Elasticsearch using the following commands:

  sudo apt update

  sudo apt install elasticsearch

Configuration:

You can configure Elasticsearch by modifying settings within the */etc*/elasticsearch/*elasticsearch.yml* file.

1. Locate the Elasticsearch Configuration File:

- Navigate to the directory containing the Elasticsearch configuration file */etc/elasticsearch/*.

2. Edit the *elasticsearch.yml* File:

- Open the *elasticsearch.yml* file in a text editor with appropriate permissions.

3. Configure HTTP and Transport Hosts:

- Set the *http.host* and *transport.host* parameters to *127.0.0.1* or the desired IP address.

4. Configure Cluster Name:

- Set the *cluster.name* parameter to the desired name. This name helps identify the Elasticsearch cluster.

5. Configure Thread Pool Search Queue Size:

- Set the *thread_pool.search.queue_size* parameter to the desired value, such as *100000*.

6. Configure Paths for Logs and Data:

- Set the *path.logs* and *path.data* parameters to the desired directories, such as "*/var/log/elasticsearch*" and "*/var/lib/elasticsearch*", respectively.

7. Configure X-Pack Security (Optional):

- If you're not using X-Pack security, ensure that *xpack.security.enabled* is set to *false*.

8. Configure Script Allowed Types (Optional):

- If needed, set the *script.allowed_types* parameter to specify allowed script types.

9. Save the Changes:

- After making the necessary configurations, save the changes to the *elasticsearch.yml* file.

10. Custom JVM Options:

- Create the file */etc/elasticsearch/jvm.options.d/jvm.options* if it doesn't exist.

11. Custom JVM Options:

- Inside *jvm.options*, add the desired JVM options, such as:
    - -Dlog4j2.formatMsgNoLookups=true
    - -Xms4g
    - -Xmx4g

- Adjust the memory settings (-Xms and -Xmx) according to the available memory.

/etc/elasticsearch/elasticsearch.yml

```
http.host: 127.0.0.1
transport.host: 127.0.0.1
cluster.name: hive
thread_pool.search.queue_size: 100000
path.logs: "/var/log/elasticsearch"
path.data: "/var/lib/elasticsearch"
xpack.security.enabled: false
script.allowed_types: "inline,stored"
```

NOTE:

Index creation occurs during TheHive's initial startup, which may take some time to complete.

Similar to data and files, indexes should be included in the backup policy to ensure their preservation.

Indexes can be removed and re-created as needed.

Start the service:

1.Start the Service:

- Execute the following command to start the Elasticsearch service:

sudo systemctl start elasticsearch

2.Ensure Service Restarts After Reboot:

- Enable the Elasticsearch service to restart automatically after a system reboot:

sudo systemctl enable elasticsearch

3.(Optional) Remove Existing Data Before Starting:

- If the Elasticsearch service was started automatically before configuring it, it's recommended to stop it, remove existing data, and restart it once the configuration is updated. Execute the following commands:

sudo systemctl stop elasticsearch

*sudo rm -rf /var/lib/elasticsearch/**

File Storage:

For standalone production and test servers, we recommend using the local filesystem.

- To utilize the local filesystem for file storage, begin by selecting a dedicated folder. By default, this folder is located at */opt/thp/thehive/files*:

  sudo mkdir -p /opt/thp/thehive/files

- This path will be utilized in the configuration of TheHive. After installing TheHive, it's important to ensure that the user thehive owns the chosen path for storing files:

  chown -R thehive:thehive /opt/thp/thehive/files

TheHive:

Installation:

- For Debian systems, use the following commands:

  wget -O- https://raw.githubusercontent.com/StrangeBeeCorp/Security/main/PGP%20keys/packages.key | sudo gpg --dearmor -o /usr/sh

- Install TheHive package by using the following commands:

  echo 'deb [arch=all signed-by=/usr/share/keyrings/strangebee-archive-keyring.gpg] https://deb.strangebee.com thehive-5.4 main' |sudo tee -a /etc/apt/sources.list.d/strangebee.list

  sudo apt-get update

  sudo apt-get install -y thehive

Configuration:

The setup provided with binary packages is tailored for a standalone installation, with all components hosted on the same server. At this point, it's crucial to fine-tune the following parameters as necessary:

/etc/thehive/application.conf

```
[..]
# Service configuration
application.baseUrl = "http://localhost:9000" #
play.http.context = "/"                       #
[..]
```

The following configurations are necessary for successful initiation of TheHive:

- Secret key configuration
- Database configuration
- File storage configuration

Secret key configuration:

The secret key is automatically generated and stored in */etc/thehive/secret.conf* during package installation.

Database & index:

By default, TheHive is configured to connect to local Cassandra and Elasticsearch databases.

/etc/thehive/application.conf

```
# Database and index configuration
# By default, TheHive is configured to connect to local Cassandra 4.x and a
# local Elasticsearch services without authentication.
db.janusgraph {
storage {
    backend = cql
    hostname = ["127.0.0.1"]
    # Cassandra authentication (if configured)
    # username = "thehive"
    # password = "password"
    cql {
    cluster-name = thp
    keyspace = thehive
    }
}
index.search {
    backend = elasticsearch
    hostname = ["127.0.0.1"]
    index-name = thehive
}
}
```

File storage:

The default file storage location of TheHive is */opt/thp/thehive/files*.

If you decide to store files on the local filesystem:

1. Ensure thehive user has permissions on the destination folder:

chown -R thehive:thehive /opt/thp/thehive/files

2. Default values in the configuration file:

/etc/thehive/application.conf

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

Run:

- To start TheHive service and enable it to run on system boot, execute the following commands in your terminal:

  sudo systemctl start thehive

  sudo systemctl enable thehive

- After the service has successfully started, launch your web browser and navigate to *http://YOUR_SERVER_ADDRESS:9000/*
- The default admin user credentials are as follows:

  Username: admin@thehive.local

  Password: secret

CORTEX

RUN THE CMD LINE IN TERMINAL:

sudo wget https://github.com/TheHive-Project/Cortex/releases/download/3.1.8/cortex_3.1.8-1_all.deb

sudo dpkg -i cortex_3.1.8-1_all.deb

sudo apt -f install (optional-only use in the place of missing packages)

MISP

RUN THE CMD LINE IN TERMINAL:

sudo wget
https://raw.githubusercontent.com/MISP/MISP/refs/heads/2.5/INSTALL/INSTALL.ubuntu2404.sh

chmod +x INSTALL.ubuntu2404.sh

./INSTALL.ubuntu2404.sh

KIBANA

PREREQUISITES:

- Before installing Kibana, ensure that you have the necessary dependencies, including Node.js and Elasticsearch. Ensure Elasticsearch is running and accessible.

sudo apt-get update

sudo apt-get install -y wget

Install Kibana:

- Go to the Kibana downloads page and find the version that matches your Elasticsearch version. Use wget to download it.

wget https://artifacts.elastic.co/downloads/kibana/7.17.12/kibana-7.17.12-amd64.deb

- Use the dpkg command to install the downloaded package.

sudo dpkg -i kibana-7.17.12-amd64.deb

Configure Kibana:

1. Edit the Kibana Configuration File:
- Open the kibana.yml configuration file, usually located in /etc/kibana.

sudo nano /etc/kibana/kibana.yml

2. Set the Elasticsearch URL:

- Uncomment and set the Elasticsearch URL. Replace localhost with the IP address of your Elasticsearch server if it's not on the same machine.

elasticsearch.hosts: ["http://localhost:9200"]

3.Set the Server Host and Port:

- You can set the server host and port as needed. For local access, you might set:

server.host: "0.0.0.0"

server.port: 5601

Start Kibana:

1.Start the Kibana service using the following command:

sudo systemctl start kibana

2.Enable Kibana on Boot:

- To ensure Kibana starts automatically on boot, enable the service:

sudo systemctl enable kibana

Access Kibana:

- Open your web browser and go to *http://<your_kibana_ip>:5601*. You should see the Kibana interface.

Connect Kibana with TheHive:

To connect Kibana with TheHive, you need to configure TheHive to send logs to Elasticsearch, which Kibana will visualize.

1. Configure TheHive to Send Logs to Elasticsearch:
- Make sure you have the correct settings in TheHive's configuration to send logs to your Elasticsearch instance.
2. Access Kibana:
- After ensuring that logs from TheHive are being sent to Elasticsearch, you can create index patterns in Kibana to visualize TheHive data.
    - Go to the Management section in Kibana.
    - Click on Index Patterns.
    - Create a new index pattern that matches TheHive logs (e.g., *thehive-*).
3. Visualize TheHive Data:
- Once you have created the index pattern, you can start creating visualizations and dashboards using TheHive's data.
