Step-by-Step guide to setting up Elastic Security Stack with Fleet Server, AI/ML for threat hunting, Open-Source Sandbox, and TheHive SOAR

Applies to the latest edition of Ubuntu Server. This guide assumes you have a fresh installation of Ubuntu Server and basic knowledge of Linux commands.

**Prepare Ubuntu Server**

1. **Update System Packages:**

sudo apt update

sudo apt upgrade -y

1. **Install Necessary Packages:**

sudo apt install -y curl gnupg software-properties-common

**2. Install Elastic Stack (Elasticsearch, Kibana, and Fleet Server)**

1. **Add Elastic Stack APT Repository:**

wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -

sudo sh -c 'echo "deb https://artifacts.elastic.co/packages/8.x/apt stable main" > /etc/apt/sources.list.d/elastic-8.x.list'

sudo apt update

1. **Install Elasticsearch:**

sudo apt install -y elasticsearch

**Configure Elasticsearch:** Edit the Elasticsearch configuration file:

sudo nano /etc/elasticsearch/elasticsearch.yml

Set the following configurations (adjust as needed):

network.host: 0.0.0.0

discovery.type: single-node

**Start and enable Elasticsearch:**

sudo systemctl start elasticsearch

sudo systemctl enable elasticsearch

1. **Install Kibana:**

sudo apt install -y kibana

**Configure Kibana:** Edit the Kibana configuration file:

sudo nano /etc/kibana/kibana.yml

Set the following configurations:

server.host: "0.0.0.0"

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

Start and enable Kibana:

sudo systemctl start kibana

sudo systemctl enable kibana

1. **Install Fleet Server:**

sudo apt install -y elastic-agent

1. **Configure Fleet Server:** You will need to enroll the Fleet Server with Kibana. Follow the instructions in the Kibana UI or the official documentation for the enrollment process.

**3. Configure AI/ML for Threat Hunting**

1. **Enable Machine Learning in Kibana:**
   * Open Kibana in your browser: http://<your-server-ip>:5601
   * Go to Management > Machine Learning > Data Frames.
   * Create and configure your machine learning jobs for threat detection as per your requirements.
2. **Set Up Anomaly Detection Jobs:**
   * Navigate to Machine Learning > Anomaly Detection.
   * Create jobs based on your data and the types of threats you want to detect.

**4. Set Up Open Source Sandbox**

1. **Install Cuckoo Sandbox:**

sudo apt install -y python3-pip

sudo pip3 install cuckoo

**Configure Cuckoo:**

cuckoo init

sudo nano /etc/cuckoo/cuckoo.conf

Adjust configurations for network, analysis, etc.

1. **Start Cuckoo Sandbox:**

sudo systemctl start cuckoo

sudo systemctl enable cuckoo

Access Cuckoo's web interface as per the configuration in your “cuckoo.conf”.

1. **Install and Configure TheHive SOAR**
2. **Add TheHive Repository:**

sudo wget -O - https://deb.thehive-project.org/public.gpg | sudo apt-key add -

sudo sh -c 'echo "deb https://deb.thehive-project.org stable main" > /etc/apt/sources.list.d/thehive.list'

sudo apt update

1. **Install TheHive:**

sudo apt install -y thehive

**Configure TheHive:** Edit the configuration file:

sudo nano /etc/thehive/application.conf

Set the necessary configurations, such as database connection and HTTP settings.

Start and enable TheHive:

sudo systemctl start thehive

sudo systemctl enable thehive

1. **Install and Configure Cortex (for TheHive integrations):**

sudo wget -O - https://deb.cortex.project.org/public.gpg | sudo apt-key add -

sudo sh -c 'echo "deb https://deb.cortex.project.org stable main" > /etc/apt/sources.list.d/cortex.list'

sudo apt update

sudo apt install -y cortex

**Configure Cortex:** Edit the configuration file:

sudo nano /etc/cortex/application.conf

Start and enable Cortex:

sudo systemctl start cortex

sudo systemctl enable cortex

**6. Integrate All Components**

1. **Integrate Elasticsearch with Kibana: Ensure Kibana is connected to Elasticsearch. Access Kibana through your browser and verify data ingestion.**
2. **Integrate Fleet Server with Elasticsearch: Follow the Kibana UI instructions to finalize Fleet Server integration.**
3. **Connect TheHive with Elasticsearch: Update TheHive configuration to connect to your Elasticsearch instance.**
4. **Integrate Cuckoo Sandbox with TheHive: Configure Cuckoo to send alerts to TheHive using available integrations or API.**

**1. Integrate Elasticsearch with Kibana**

1. **Check Elasticsearch Status:** Ensure Elasticsearch is running and accessible:

curl -X GET "localhost:9200/"

1. **Configure Kibana to Connect to Elasticsearch:** Edit the Kibana configuration file:

sudo nano /etc/kibana/kibana.yml

Set the elasticsearch.hosts property to point to your Elasticsearch instance:

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

Save and exit the file.

1. **Restart Kibana:**

sudo systemctl restart kibana

1. **Verify Connection:** Open Kibana in your browser: http://<your-server-ip>:5601 Go to Management > Kibana > Index Patterns and create an index pattern to verify Kibana can access Elasticsearch data.

**2. Integrate Fleet Server with Elasticsearch**

1. **Configure Fleet Server in Kibana:**
   * Open Kibana in your browser.
   * Navigate to Management > Fleet.
   * Follow the prompts to enroll a Fleet Server. Kibana will provide you with a command to run on your Fleet Server to complete the enrollment.
2. **Run Enrollment Command on Fleet Server:** Copy and execute the command provided by Kibana on your Fleet Server machine. This command typically looks like:

sudo elastic-agent enroll \

--url=http://<your-kibana-ip>:5601 \

--fleet-server-es=http://localhost:9200 \

--fleet-server-service-token=<token> \

--fleet-server-policy=<policy-id>

**2. Verify Fleet Server Enrollment:** In Kibana, go to Fleet > Agents and check if the Fleet Server appears and is healthy.

**3. Connect TheHive with Elasticsearch**

1. **Update TheHive Configuration:** Edit TheHive’s configuration file:

sudo nano /etc/thehive/application.conf

Find and update the elasticsearch section to point to your Elasticsearch instance:

elasticsearch {

nodes = ["http://localhost:9200"]

}

Save and exit the file.

1. **Restart TheHive:**

sudo systemctl restart thehive

1. **Verify Integration:** Open TheHive in your browser and ensure you can create and manage cases. Verify that TheHive can index and search data in Elasticsearch.

**4. Integrate Cortex with TheHive**

1. **Update Cortex Configuration:** Edit Cortex’s configuration file:

sudo nano /etc/cortex/application.conf

Set the thehive section to point to your TheHive instance:

thehive {

url = "http://localhost:9000"

api-key = "your-thehive-api-key"

}

Save and exit the file.

1. **Restart Cortex:**

sudo systemctl restart cortex

1. **Verify Integration:** In TheHive, go to Settings > Integrations and ensure Cortex is listed and configured. You should be able to create observables and have Cortex analyze them.

**5. Integrate Cuckoo Sandbox with TheHive**

1. **Configure Cuckoo to Send Alerts to TheHive:**
   * Edit Cuckoo’s configuration file:

sudo nano /etc/cuckoo/cuckoo.conf

Locate the section for integration and add your TheHive instance details:

[thehive]

url = http://localhost:9000

api\_key = your-thehive-api-key

Save and exit the file.

1. **Restart Cuckoo:**

sudo systemctl restart cuckoo

1. **Verify Integration:**
   * Submit a sample file to Cuckoo for analysis.
   * Check TheHive for new alerts or cases created by Cuckoo.

**6. Verify Overall Integration**

1. **Test Data Flow:**
   * Submit sample data to Fleet and verify that it appears in Kibana.
   * Run a machine learning job in Kibana and check for alerts or anomalies.
   * Create a case in TheHive and verify that the alerts or data are reflected.
   * Analyze a sample file with Cuckoo and ensure results are properly logged and linked to TheHive.
2. **Monitor Logs:**
   * Check logs for Elasticsearch, Kibana, TheHive, Cortex, and Cuckoo for any errors or issues:

sudo journalctl -u <service-name>

1. **Regular Updates and Maintenance:**

* Regularly update all components to ensure security and compatibility.
* Periodically review and adjust configurations as needed.

**7. Verify and Test**

1. **Access Kibana and Verify Data:** Open Kibana and check that data is being ingested and visualized correctly.
2. **Test Machine Learning Jobs:** Ensure that machine learning jobs are running and detecting anomalies as expected.
3. **Test Cuckoo Sandbox:** Submit test samples to Cuckoo and verify that they are processed and analyzed correctly.
4. **Verify TheHive Functionality:** Access TheHive’s web interface and create/test cases to ensure it’s working correctly with integrated components.

**Additional Notes**

* Always refer to official documentation for specific version details and advanced configurations.
* Regularly update all components to ensure security and compatibility.