Security Operations Center

Documentatie

Cybersecurity & Security Operation Center

Ferdi 3CCS02

Academiejaar 2022-2023

Campus Geel, Kleinhoefstraat 4, BE-2440 Geel

LID VAN ASSOCIATIE KU LEUVEN

THOMAS MORE

INHOUDSTAFEL

Inhoud

1 INLEIDING

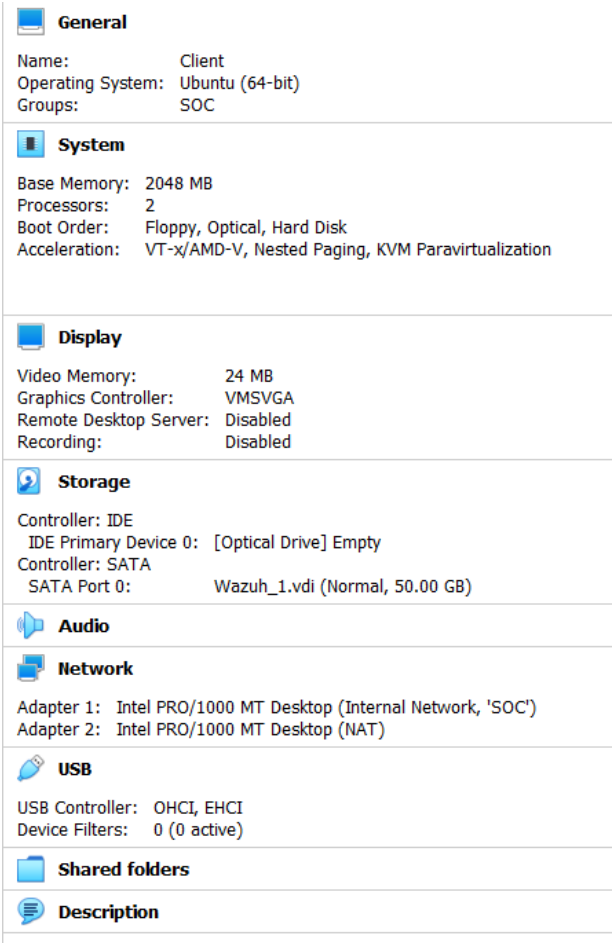
We hebben de opdracht gekregen om een eigen Security Operation Center Software Stack op te zetten om een productiesysteem te beveiligen. Hiervoor moesten we een systeem opzetten naar keuze en op dat systeem een data gaan capteren en collecteren en deze door sturen naar alerts in een SIEM. Deze SIEM gaat dan een alert triggeren op en SOAR-systeem en eventueel een Incident Response systeem.

Dan moesten we een aanval doen op het productiesysteem, die aanval gaat data genereren in de collector. Vervolgens gaat de SIEM een alarm triggeren die de SOAR in actie zet om te herstellen.

2 PRODUCTION

Voor het productiesysteem heb ik gekozen voor een Ubuntu 22.04 desktop machine opgezet via virtualbox.

Hierbij heb ik eerst in virtualbox een nieuwe machine aangemaakt met onderstaande settings:



General

Name: Client
Operating System: Ubuntu (64-bit)
Groups: SOC

System

Base Memory: 2048 MB
Processors: 2
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization

Display

Video Memory: 24 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage

Controller: IDE
IDE Primary Device 0: [Optical Drive] Empty
Controller: SATA
SATA Port 0: Wazuh_1.vdi (Normal, 50.00 GB)

Audio

Network

Adapter 1: Intel PRO/1000 MT Desktop (Internal Network, 'SOC')
Adapter 2: Intel PRO/1000 MT Desktop (NAT)

USB

USB Controller: OHCI, EHCI
Device Filters: 0 (0 active)

Shared folders

Description

3 COLLECTION & DETECTION

Voor Collection & Detection heb ik ervoor gekozen om Suricata te gebruiken dit heb ik op volgende manier geïnstalleerd.

Stap 1 Begin:

In een terminal een sudo apt update && sudo apt upgrade te doen.

Daarna de verschillende dependencies installeren met volgend commando:

"apt-get install build-essential libpcap-dev \

libnet1-dev libyaml-0-2 libyaml-dev pkg-config zlib1g zlib1g-dev \

libcap-ng-dev libcap-ng0 make libmagic-dev \

libgeoip-dev liblua5.1-dev libhiredis-dev libevent-dev \

python-yaml rustc cargo libpcre2-dev"

Stap 2 installeren van Suricata:

Met volgende commando's kan je Suricata installeren.

"sudo apt-get install software-properties-common

sudo add-apt-repository ppa:oisf/suricata-stable

sudo apt-get update

sudo apt-get install suricata"

Source: https://suricata.readthedocs.io/en/latest/install.html

4 SIEM

Voor de SIEM heb ik gekozen om met Wazuh te werken. Je hebt hier een OVA-file van die je kan gebruiken maar wegens problemen ben ik uiteindelijk overgegaan op Ubuntu Server 22.04 LTS. Na de installatie van de server heb ik de documentatie gevolgd van Wazuh om de manager en de agent te installeren.

4.1 Ubuntu Server/ Wazuh Manager:

4.1.1 Wazuh Indexer

Commando's:

curl -sO https://packages.wazuh.com/4.3/wazuh-certs-tool.sh

curl -sO https://packages.wazuh.com/4.3/config.yml

Daarna de config.yml aanpassen:

```
nodes:
  # Wazuh indexer nodes
  indexer:
    - name: soc_indexer
      ip: 10.0.3.2
    #- name: node-2
    #  ip: <indexer-node-ip>
    #- name: node-3
    #  ip: <indexer-node-ip>

  # Wazuh server nodes
  # If there is more than one Wazuh server
  # node, each one must have a node_type
  server:
    - name: soc_server
      ip: 10.0.3.2
    #  node_type: master
    #- name: wazuh-2
    #  ip: <wazuh-manager-ip>
    #  node_type: worker
    #- name: wazuh-3
    #  ip: <wazuh-manager-ip>
    #  node_type: worker

  # Wazuh dashboard nodes
  dashboard:
    - name: soc_dashboard
      ip: 10.0.3.2
```

Dan volgende commando's uitvoeren:

bash ./wazuh-certs-tool.sh -A

tar -cvf ./wazuh-certificates.tar -C ./wazuh-certificates/ .

rm -rf ./wazuh-certificates

apt-get install debconf adduser procps

apt-get install gnupg apt-transport-https

curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | gpg --no-default-keyring --keyring gnupg-ring:/usr/share/keyrings/wazuh.gpg --import && chmod 644 /usr/share/keyrings/wazuh.gpg

echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable main" | tee -a /etc/apt/sources.list.d/wazuh.list

apt-get update

apt-get -y install wazuh-indexer

Configureren van Wazuh indexer door /etc/wazuh-indexer/opensearch.yml file aan te passen:

```
  GNU nano 6.2                    /etc/wazuh-indexer/opensearch.yml                    S
network.host: "10.0.3.2"
node.name: "soc_indexer"
cluster.initial_master_nodes:
- "soc_indexer"
#- "soc_dashboard"
#- "node-3"
cluster.name: "wazuh-cluster"
#discovery.seed_hosts:
#  - "node-1-ip"
#  - "node-2-ip"
#  - "node-3-ip"
node.max_local_storage_nodes: "3"
path.data: /var/lib/wazuh-indexer
path.logs: /var/log/wazuh-indexer
```

Volgende commando's:

NODE_NAME= soc_indexer

mkdir /etc/wazuh-indexer/certs

tar -xf ./wazuh-certificates.tar -C /etc/wazuh-indexer/certs/ ./$NODE_NAME.pem ./$NODE_NAME-key.pem ./admin.pem ./admin-key.pem ./root-ca.pem

mv -n /etc/wazuh-indexer/certs/$NODE_NAME.pem /etc/wazuh-indexer/certs/indexer.pem

mv -n /etc/wazuh-indexer/certs/$NODE_NAME-key.pem /etc/wazuh-indexer/certs/indexer-key.pem

chmod 500 /etc/wazuh-indexer/certs

chmod 400 /etc/wazuh-indexer/certs/*

chown -R wazuh-indexer:wazuh-indexer /etc/wazuh-indexer/certs

systemctl daemon-reload

systemctl enable wazuh-indexer

systemctl start wazuh-indexer

curl -k -u admin:admin https://<WAZUH_INDEXER_IP>:9200

4.1.2 Wazuh Server

Nu volgende de commando's om de Wazuh server te installeren:

apt-get install gnupg apt-transport-https

curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | gpg --no-default-keyring --keyring gnupg-ring:/usr/share/keyrings/wazuh.gpg --import && chmod 644 /usr/share/keyrings/wazuh.gpg

echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable main" | tee -a /etc/apt/sources.list.d/wazuh.list

apt-get update

apt-get -y install wazuh-manager

systemctl daemon-reload

systemctl enable wazuh-manager

systemctl start wazuh-manager

systemctl status wazuh-manager

apt-get -y install filebeat

curl -so /etc/filebeat/filebeat.yml https://packages.wazuh.com/4.3/tpl/wazuh/filebeat/filebeat.yml

Vervolgens moet je de etc/filebeat/filebeat.yml file aanpassen naar je gebruikte ip adres:

```
  GNU nano 6.2                          /etc/filebeat/filebeat.yml
# Wazuh - Filebeat configuration file
output.elasticsearch:
  hosts: ["10.0.3.2:9200"]
  protocol: https
  username: ${username}
  password: ${password}
  ssl.certificate_authorities:
    - /etc/filebeat/certs/root-ca.pem
  ssl.certificate: "/etc/filebeat/certs/filebeat.pem"
  ssl.key: "/etc/filebeat/certs/filebeat-key.pem"
setup.template.json.enabled: true
setup.template.json.path: '/etc/filebeat/wazuh-template.json'
setup.template.json.name: 'wazuh'
setup.ilm.overwrite: true
setup.ilm.enabled: false
```

Daarna volgende commando's:

filebeat keystore create

echo admin | filebeat keystore add username --stdin --force

echo admin | filebeat keystore add password --stdin –force

```
curl -so /etc/filebeat/wazuh-template.json
https://raw.githubusercontent.com/wazuh/wazuh/4.3/extensions/elastics
earch/7.x/wazuh-template.json

chmod go+r /etc/filebeat/wazuh-template.json

curl -s https://packages.wazuh.com/4.x/filebeat/wazuh-filebeat-
0.2.tar.gz | tar -xvz -C /usr/share/filebeat/module

NODE_NAME=soc_server

mkdir /etc/filebeat/certs

tar -xf ./wazuh-certificates.tar -C /etc/filebeat/certs/
./$NODE_NAME.pem ./$NODE_NAME-key.pem ./root-ca.pem

mv -n /etc/filebeat/certs/$NODE_NAME.pem
/etc/filebeat/certs/filebeat.pem

mv -n /etc/filebeat/certs/$NODE_NAME-key.pem
/etc/filebeat/certs/filebeat-key.pem

chmod 500 /etc/filebeat/certs

chmod 400 /etc/filebeat/certs/*

chown -R root:root /etc/filebeat/certs

systemctl daemon-reload

systemctl enable filebeat

systemctl start filebeat

filebeat test output
```

4.1.3 Wazuh Dashboard

Als laatste moet je het Dashboard installeren op volgende manier:

```
apt-get install debhelper tar curl libcap2-bin

apt-get install gnupg apt-transport-https

curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | gpg --no-
default-keyring --keyring gnupg-ring:/usr/share/keyrings/wazuh.gpg --
import && chmod 644 /usr/share/keyrings/wazuh.gpg

echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg]
https://packages.wazuh.com/4.x/apt/ stable main" | tee -a
/etc/apt/sources.list.d/wazuh.list

apt-get update

apt-get -y install wazuh-dashboard
```

De yml file Aanpassen om het juiste ip adres te gebruiken

```
  GNU nano 6.2                /etc/wazuh-dashboard/opensearch_dashboards.yml
server.host: 10.0.3.2
server.port: 443
opensearch.hosts: https://10.0.3.2:9200
opensearch.ssl.verificationMode: certificate
#opensearch.username:
#opensearch.password:
opensearch.requestHeadersWhitelist: ["securitytenant","Authorization"]
opensearch_security.multitenancy.enabled: false
opensearch_security.readonly_mode.roles: ["kibana_read_only"]
server.ssl.enabled: true
server.ssl.key: "/etc/wazuh-dashboard/certs/dashboard-key.pem"
server.ssl.certificate: "/etc/wazuh-dashboard/certs/dashboard.pem"
opensearch.ssl.certificateAuthorities: ["/etc/wazuh-dashboard/certs/root-ca.pem"]
uiSettings.overrides.defaultRoute: /app/wazuh
```

NODE_NAME=soc_dashboard

mkdir /etc/wazuh-dashboard/certs

tar -xf ./wazuh-certificates.tar -C /etc/wazuh-dashboard/certs/ ./$NODE_NAME.pem ./$NODE_NAME-key.pem ./root-ca.pem

mv -n /etc/wazuh-dashboard/certs/$NODE_NAME.pem /etc/wazuh-dashboard/certs/dashboard.pem

mv -n /etc/wazuh-dashboard/certs/$NODE_NAME-key.pem /etc/wazuh-dashboard/certs/dashboard-key.pem

chmod 500 /etc/wazuh-dashboard/certs

chmod 400 /etc/wazuh-dashboard/certs/*

chown -R wazuh-dashboard:wazuh-dashboard /etc/wazuh-dashboard/certs

systemctl daemon-reload

systemctl enable wazuh-dashboard

systemctl start wazuh-dashboard

4.1.4 Wazuh Agent

Op de Ubuntu client moet je ook de Wazuh Agent installeren

curl -so wazuh-agent-4.3.10.deb https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.3.10-1_amd64.deb && sudo WAZUH_MANAGER='10.0.3.2' WAZUH_AGENT_GROUP='default' dpkg -i ./wazuh-agent-4.3.10.deb

sudo systemctl daemon-reload

sudo systemctl enable wazuh-agent

sudo systemctl start wazuh-agent

5 AUTOMATION SOAR

Voor de SOAR heb ik gekozen om Shuffle te gebruiken dit kan je op volgende manier installeren op een Ubuntu server.

```
git clone https://github.com/Shuffle/Shuffle

cd Shuffle

mkdir shuffle-database

sudo chown -R 1000:1000 shuffle-database

docker-compose up -d
```

6 INCIDENT RESPONSE

Voor Incident Response heb ik TheHive met integratie van Cortex

6.1 The Hive

6.1.1 Java Virtual Machine

```
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"
```

6.1.2 Apache Cassandra

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

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

sudo apt update

sudo apt install Cassandra
```

Configuratie van Cassandra gebeurt in de /etc/cassandra/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'
[..]
```

sudo systemctl start cassandra

6.1.3 Elasticsearch

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

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

sudo apt update

sudo apt install elasticsearch

Configuratie gebeurt in /etc/elasticsearch/elasticsearch.yml file

```
  GNU nano 6.2                    /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"
```

sudo systemctl start elasticsearch

6.1.4 File Storage

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

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

6.1.5 The Hive 5

wget -O- https://archives.strangebee.com/keys/strangebee.gpg | sudo gpg --dearmor -o /usr/share/keyrings/strangebee-archive-keyring.gpg

echo 'deb [signed-by=/usr/share/keyrings/strangebee-archive-keyring.gpg] https://deb.strangebee.com thehive-5.x main' | sudo tee -a /etc/apt/sources.list.d/strangebee.list

sudo apt-get update

sudo apt-get install -y thehive

configuratie van application.conf file

```
  GNU nano 6.2                                                    /etc/thehive/application.conf *
#   play.http.secret.key="$(cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 64 |#   head -n 1)"
#   _EOF_
include "/etc/thehive/secret.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 = ["10.0.3.4"]
    # 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
  }
}

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

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

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

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

sudo systemctl start thehive

sudo systemctl enable thehive

6.2 Cortex

The Hive heeft en integratie met cortex en daarom heb ik deze er ook bij ingezet.

wget -O- "https://raw.githubusercontent.com/TheHive-Project/Cortex/master/PGP-PUBLIC-KEY" | sudo apt-key add -

wget -qO- https://raw.githubusercontent.com/TheHive-Project/Cortex/master/PGP-PUBLIC-KEY | sudo gpg --dearmor -o /usr/share/keyrings/thehive-project.gpg

```
echo 'deb https://deb.thehive-project.org release main' | sudo tee -a
/etc/apt/sources.list.d/thehive-project.list

apt install cortex
```

6.2.1 Secret Key Configuration

In de file /etc/cortex/application.conf bijzetten :

```
include /etc/cortex/secret.conf
```

6.2.2 Analyzers & Responders

Commando's:

```
sudo apt install -y --no-install-recommends python3-pip python3-
dev ssdeep libfuzzy-dev libfuzzy2 libimage-exiftool-perl libmagic1
build-essential git libssl-dev

sudo pip3 install -U pip setuptools

cd /opt

git clone https://github.com/TheHive-Project/Cortex-Analyzers

chown -R cortex:cortex /opt/Cortex-Analyzers

cd /opt

for I in $(find Cortex-Analyzers -name 'requirements.txt'); do
sudo -H pip3 install -r $I || true; done
```

```
systemctl start cortex
```

General settings

Server name

Soc Cortex

Server url *

http://10.0.3.4:9001

API Key *

.....

Proxy

Use default configuration | Enabled | Disabled

SSL Settings

Do not check Certificate Autority

Not recommended

Disable hostname Verification

Advanced settings

Choose the filter on TheHive organisations

Include all organisations

7 THREAT INTEL

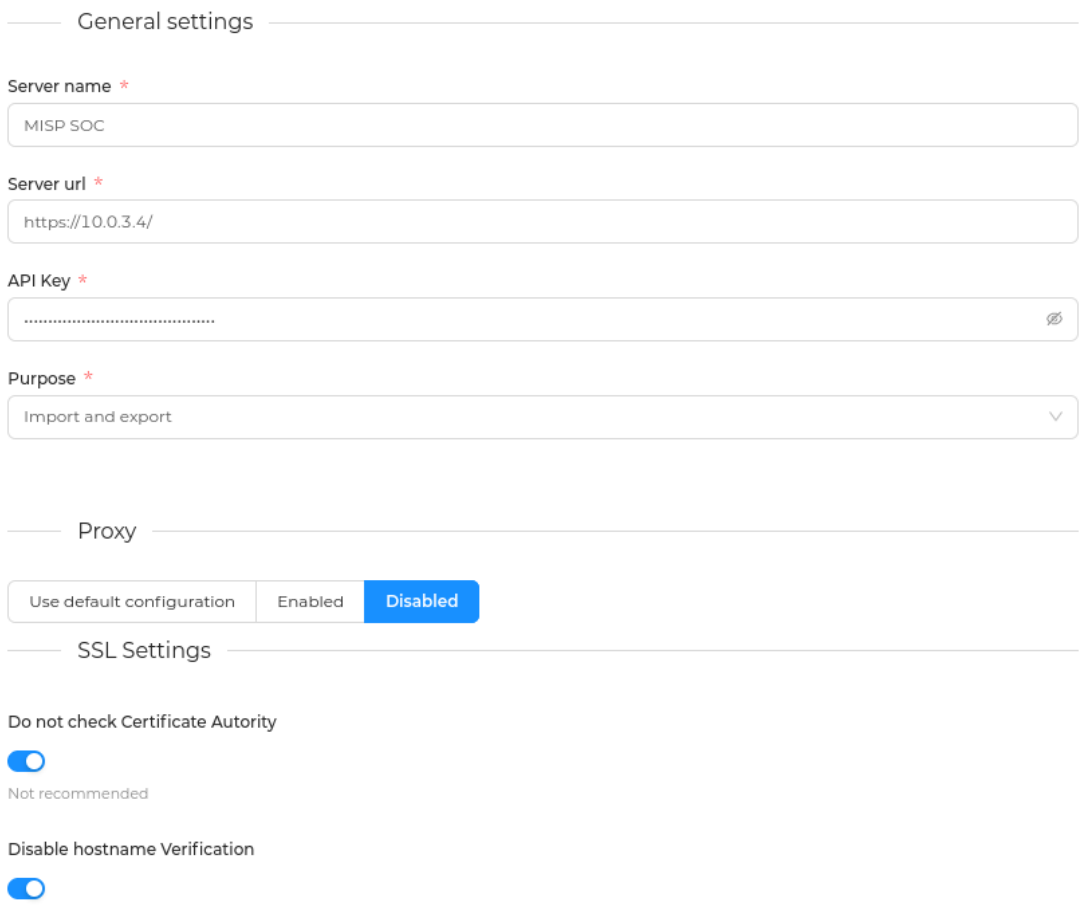
Voor Threat Intel heb ik gekozen om de MISP-integratie bij The Hive te gebruiken.

Commando's:

wget -O /tmp/INSTALL.sh https://raw.githubusercontent.com/MISP/MISP/2.4/INSTALL/INSTALL.sh

bash /tmp/INSTALL.sh

met optie -A van install all tijdens het vragen.



8 CONCLUSIE

Ik heb de taak niet volledig kunnen maken, ik heb wel alle aparte onderdelen kunnen laten werken maar heb de automatisatie niet kunnen bekomen. Dit omdat de webhook voor Wazuh niet meewerkte en ik de oplossing niet heb kunnen vinden.