Universidad del Caribe

2000 CANCUN, QUINTANA ROO, MÉXICO

CONOCIMIENTO Y CULTURA PARA EL DESARROLLO HUMANO

INVESTIGACIÓN/REPORTE/RESUMEN:

Implementación de centro de operaciones de seguridad open source.

ASIGNATURA:

Teoría de la información

Angel Briceño Cortez

MATRÍCULA: **210300518**

PROGRAMA EDUCATIVO: **INGENIERÍA EN DATOS E INTELIGENCIA ORGANIZACIONAL**

PRESENTADO A:

PROF. Ismael Jiménez Sánchez

Preparando el ambiente de trabajo

Para esta implementación se usará una distribución de linux y mediante contenedores de docker instalaremos componentes de un centro de operaciones de seguridad tales como:

- Wazuh: Una herramienta para monitoreo y detección de amenazas.
- TheHive: Gestión de casos de incidentes e investigaciones
- Cortex:Analiza indicadores de compromisos y consultar servicios como Virustotal, Shodan, AbuseIPDB, desde una sola interfaz
- MISP: Una plataforma de inteligencia de amenazas.

Este arreglo de herramientas es útil en entornos profesionales de seguridad, todo bajo el alcance de un archivo .yaml.

El proceso de despliegue

Primero debemos asegurarnos que el sistema operativo Linux cuente con las dependencias necesarias para correr los contenedores de docker y todo lo necesario para el buen funcionamiento general del sistema operativo.

```
sudo apt update
sudo apt upgrade
```

También es recomendado instalar los paquetes de python para linux, pues algunos comandos requieren de ellos.

Una vez instalado lo anterior, se obtiene el repositorio para instalar el contenedor de wazuh en docker y generamos los certificados de nodo:

```
git clone https://github.com/wazuh/wazuh-docker.git -b v4.11.0
cd wazuh-docker/single-node
docker-compose -f generate-indexer-certs.yml run --rm generator
```

Estos son certificados SSL que aseguran la comunicación entre los componentes de Wazuh y se generan con las funciones de este.

Una vez que no hayan errores al generar los certificados, ahora podemos pasar a la creación del archivo `docker-compose.yml` con el que se levantará toda la infraestructura de seguridad:

```yaml
services:

 # TheHive, MISP, Cortex Services

 thehive:
   container_name: thehive
   image: strangebee/thehive:5.2
   restart: unless-stopped
   depends_on:
     - cassandra
     - elasticsearch
     - minio
     - cortex
   mem_limit: 1500m
   ports:
     - "0.0.0.0:9000:9000"
   environment:
     - JVM_OPTS="-Xms1024M -Xmx1024M"
   command:
     - --secret
     - "lab123456789"
     - "--cql-hostnames"
     - "cassandra"
     - "--index-backend"
     - "elasticsearch"
     - "--es-hostnames"
     - "elasticsearch"
     - "--s3-endpoint"
     - "http://minio:9002"
     - "--s3-access-key"
     - "minioadmin"
     - "--s3-secret-key"
     - "minioadmin"
     - "--s3-use-path-access-style"
   volumes:
     - ./thehive/application.conf:/etc/thehive/application.conf
     - thehivedata:/opt/thehive/data
   networks:
     - SOC_NET

 cassandra:
   container_name: cassandra
   image: 'cassandra:4'
   restart: unless-stopped
   ports:
     - "0.0.0.0:9042:9042"
   environment:
     - CASSANDRA_CLUSTER_NAME=TheHive
   volumes:
     - cassandradata:/var/lib/cassandra
   networks:
     - SOC_NET

 elasticsearch:
   container_name: elasticsearch
   image: docker.elastic.co/elasticsearch/elasticsearch:7.17.9
   restart: unless-stopped
   mem_limit: 512m
   ports:
     - "0.0.0.0:9200:9200"
   environment:
     - discovery.type=single-node
     - xpack.security.enabled=false
     - cluster.name=hive
     - http.host=0.0.0.0
     - "ES_JAVA_OPTS=-Xms256m -Xmx256m"
   volumes:
     - elasticsearchdata:/usr/share/elasticsearch/data
   networks:
     - SOC_NET

 minio:
   container_name: minio
   image: quay.io/minio/minio
   restart: unless-stopped
   command: ["minio", "server", "/data", "--console-address", ":9002"]
   environment:
     - MINIO_ROOT_USER=minioadmin
     - MINIO_ROOT_PASSWORD=minioadmin
   ports:
     - "0.0.0.0:9002:9002"
   volumes:
     - "miniodata:/data"
   networks:
     - SOC_NET

 cortex:
   container_name: cortex
   image: thehiveproject/cortex:latest
   restart: unless-stopped
   environment:
     - job_directory=/tmp/cortex-jobs
     - docker_job_directory=/tmp/cortex-jobs
   volumes:
     - /var/run/docker.sock:/var/run/docker.sock
     - /tmp/cortex-jobs:/tmp/cortex-jobs
     - ./cortex/logs:/var/log/cortex
     - ./cortex/application.conf:/cortex/application.conf
   depends_on:
     - elasticsearch
   ports:
     - "0.0.0.0:9001:9001"
   networks:
     - SOC_NET

 misp:
   container_name: misp
   image: coolacid/misp-docker:core-latest
   restart: unless-stopped
   depends_on:
     - misp_mysql
   ports:
     - "0.0.0.0:80:80"
     - "0.0.0.0:443:443"
   volumes:
     - "./server-configs/:/var/www/MISP/app/Config/"
     - "./logs/:/var/www/MISP/app/tmp/logs/"
     - "./files/:/var/www/MISP/app/files"
     - "./ssl/:/etc/nginx/certs"
   environment:
     - MYSQL_HOST=misp_mysql
     - MYSQL_DATABASE=mispdb
     - MYSQL_USER=mispuser
     - MYSQL_PASSWORD=misppass
     - MISP_BASEURL=localhost
     - TIMEZONE=Europe/London
     - "INIT=true"
     - "CRON_USER_ID=1"
     - "REDIS_FQDN=redis"
     - "HOSTNAME=https://192.168.50.200" # Update to use your actual IP
   networks:
     - SOC_NET
 misp_mysql:
   container_name: misp_mysql
   image: mysql/mysql-server:5.7
   restart: unless-stopped
   volumes:
     - mispsqldata:/var/lib/mysql
   environment:
     - MYSQL_DATABASE=mispdb
     - MYSQL_USER=mispuser
     - MYSQL_PASSWORD=misppass
     - MYSQL_ROOT_PASSWORD=mispass
   networks:
     - SOC_NET

 redis:
   container_name: redis
   image: redis:latest
   hostname: redis
   networks:
     - SOC_NET

 misp-modules:
   container_name: misp_modules
   image: coolacid/misp-docker:modules-latest
   environment:
     - "REDIS_BACKEND=redis"
   depends_on:
     - redis
     - misp_mysql
   networks:
     - SOC_NET

 # Wazuh Services

 wazuh.manager:
   image: wazuh/wazuh-manager:4.11.0
   hostname: wazuh.manager
   restart: always
   ulimits:
     memlock:
       soft: -1
       hard: -1
     nofile:
       soft: 655360
       hard: 655360
   ports:
     - "1514:1514"
     - "1515:1515"
     - "514:514/udp"
     - "55000:55000"
   environment:
     - INDEXER_URL=https://wazuh.indexer:9200
     - INDEXER_USERNAME=admin
     - INDEXER_PASSWORD=SecretPassword
     - FILEBEAT_SSL_VERIFICATION_MODE=full
     - SSL_CERTIFICATE_AUTHORITIES=/etc/ssl/root-ca.pem
     - SSL_CERTIFICATE=/etc/ssl/filebeat.pem
     - SSL_KEY=/etc/ssl/filebeat.key
     - API_USERNAME=wazuh-wui
     - API_PASSWORD=MyS3cr37P450r.*-
   volumes:
     - wazuh_api_configuration:/var/ossec/api/configuration
     - wazuh_etc:/var/ossec/etc
     - wazuh_logs:/var/ossec/logs
     - wazuh_queue:/var/ossec/queue
     - wazuh_var_multigroups:/var/ossec/var/multigroups
     - wazuh_integrations:/var/ossec/integrations
     - wazuh_active_response:/var/ossec/active-response/bin
     - wazuh_agentless:/var/ossec/agentless
     - wazuh_wodles:/var/ossec/wodles
     - filebeat_etc:/etc/filebeat
     - filebeat_var:/var/lib/filebeat
     - ./config/wazuh_indexer_ssl_certs/root-ca-manager.pem:/etc/ssl/root-ca.pem
     - ./config/wazuh_indexer_ssl_certs/wazuh.manager.pem:/etc/ssl/filebeat.pem
     - ./config/wazuh_indexer_ssl_certs/wazuh.manager-key.pem:/etc/ssl/filebeat.key
     - ./config/wazuh_cluster/wazuh_manager.conf:/wazuh-config-mount/etc/ossec.conf
   networks:
     - SOC_NET

 wazuh.indexer:
   image: wazuh/wazuh-indexer:4.11.0
   hostname: wazuh.indexer
   restart: always
   ports:
     - "9201:9200" # Changed port to avoid conflict with ElasticSearch
   environment:
     - "OPENSEARCH_JAVA_OPTS=-Xms1g -Xmx1g"
   ulimits:
     memlock:
       soft: -1
       hard: -1
     nofile:
       soft: 65536
       hard: 65536
   volumes:
     - wazuh-indexer-data:/var/lib/wazuh-indexer
     - ./config/wazuh_indexer_ssl_certs/root-ca.pem:/usr/share/wazuh-indexer/certs/root-ca.pem
     - ./config/wazuh_indexer_ssl_certs/wazuh.indexer-key.pem:/usr/share/wazuh-indexer/certs/wazuh.indexer.key
     - ./config/wazuh_indexer_ssl_certs/wazuh.indexer.pem:/usr/share/wazuh-indexer/certs/wazuh.indexer.pem
     - ./config/wazuh_indexer_ssl_certs/admin.pem:/usr/share/wazuh-indexer/certs/admin.pem
     - ./config/wazuh_indexer_ssl_certs/admin-key.pem:/usr/share/wazuh-indexer/certs/admin-key.pem
     - ./config/wazuh_indexer/wazuh.indexer.yml:/usr/share/wazuh-indexer/opensearch.yml
     - ./config/wazuh_indexer/internal_users.yml:/usr/share/wazuh-indexer/opensearch-security/internal_users.yml
   networks:
     - SOC_NET

 wazuh.dashboard:
   image: wazuh/wazuh-dashboard:4.11.0
   hostname: wazuh.dashboard
   restart: always
   ports:
     - "8443:5601" # Changed port to 8443 to avoid conflict with MISP
   environment:
     - INDEXER_USERNAME=admin
     - INDEXER_PASSWORD=SecretPassword
     - WAZUH_API_URL=https://wazuh.manager
     - DASHBOARD_USERNAME=kibanaserver
     - DASHBOARD_PASSWORD=kibanaserver
```

```
     - API_USERNAME=wazuh-wui
     - API_PASSWORD=MyS3cr37P450r.*-
   volumes:
     -
./config/wazuh_indexer_ssl_certs/wazuh.dashboard.pem:/usr/share/wazuh-dash
board/certs/wazuh-dashboard.pem
     -
./config/wazuh_indexer_ssl_certs/wazuh.dashboard-key.pem:/usr/share/wazuh-
dashboard/certs/wazuh-dashboard-key.pem
     -
./config/wazuh_indexer_ssl_certs/root-ca.pem:/usr/share/wazuh-dashboard/ce
rts/root-ca.pem
     -
./config/wazuh_dashboard/opensearch_dashboards.yml:/usr/share/wazuh-dashbo
ard/config/opensearch_dashboards.yml
     -
./config/wazuh_dashboard/wazuh.yml:/usr/share/wazuh-dashboard/data/wazuh/c
onfig/wazuh.yml
     - wazuh-dashboard-config:/usr/share/wazuh-dashboard/data/wazuh/config
     -
wazuh-dashboard-custom:/usr/share/wazuh-dashboard/plugins/wazuh/public/ass
ets/custom
   depends_on:
     - wazuh.indexer
   networks:
     - SOC_NET

volumes:
 # TheHive/MISP/Cortex volumes
 miniodata:
 cassandradata:
 elasticsearchdata:
 thehivedata:
 mispsqldata:

 # Wazuh volumes
 wazuh_api_configuration:
 wazuh_etc:
 wazuh_logs:
 wazuh_queue:
 wazuh_var_multigroups:
 wazuh_integrations:
 wazuh_active_response:
 wazuh_agentless:
 wazuh_wodles:
 filebeat_etc:
```

```
 filebeat_var:
 wazuh-indexer-data:
 wazuh-dashboard-config:
 wazuh-dashboard-custom:

networks:
 SOC_NET:
   driver: bridge
```

Ahora bien, a todo esto, que es exactamente un archivo .yml o .yaml?
Se trata de archivos de configuración e intercambio de datos entre lenguajes de programación y hecho de forma que sea más fácil de escribir para los humanos. Es más común trabajar este tipo de archivos en ciberseguridad y DevOps ya sea para:

- Levantar contenedores (como en este caso).
- Definir despliegues y servicios en Kubernetes
- Aplicaciones de CI/CD en GitHub Actions y GitLab
- Configuración de servidores y analizadores

Por ende, el código de nuestro `docker-compose.yml` está hecho para que dentro de un solo archivo, se declaran variables, configuraciones y establecer un estructura dentro de un código que es más fácil de entender en lugar de utilizar algún otro lenguaje de programación.
Así mismo, se debe crear un archivo de configuración aparte para nuestro servicio de cortex y thehive:

```
mkdir -p cortex thehive && touch cortex/application.conf
thehive/application.conf
sudo nano cortex/application.conf
```

Y dentro de `application.conf` añadimos el siguiente código:

```
# Secret key
play.http.secret.key="5jU6h1euT1jMJt3uCe3fbO2iGcoXOkF97XESPxkALivHblLd3vw8
Vh4rJYpfL2wXcc"

# HTTP configuration
http.address=0.0.0.0
http.port=9001

# Akka configuration
akka {
 cluster.enable = off
 actor {
   provider = local
 }
```

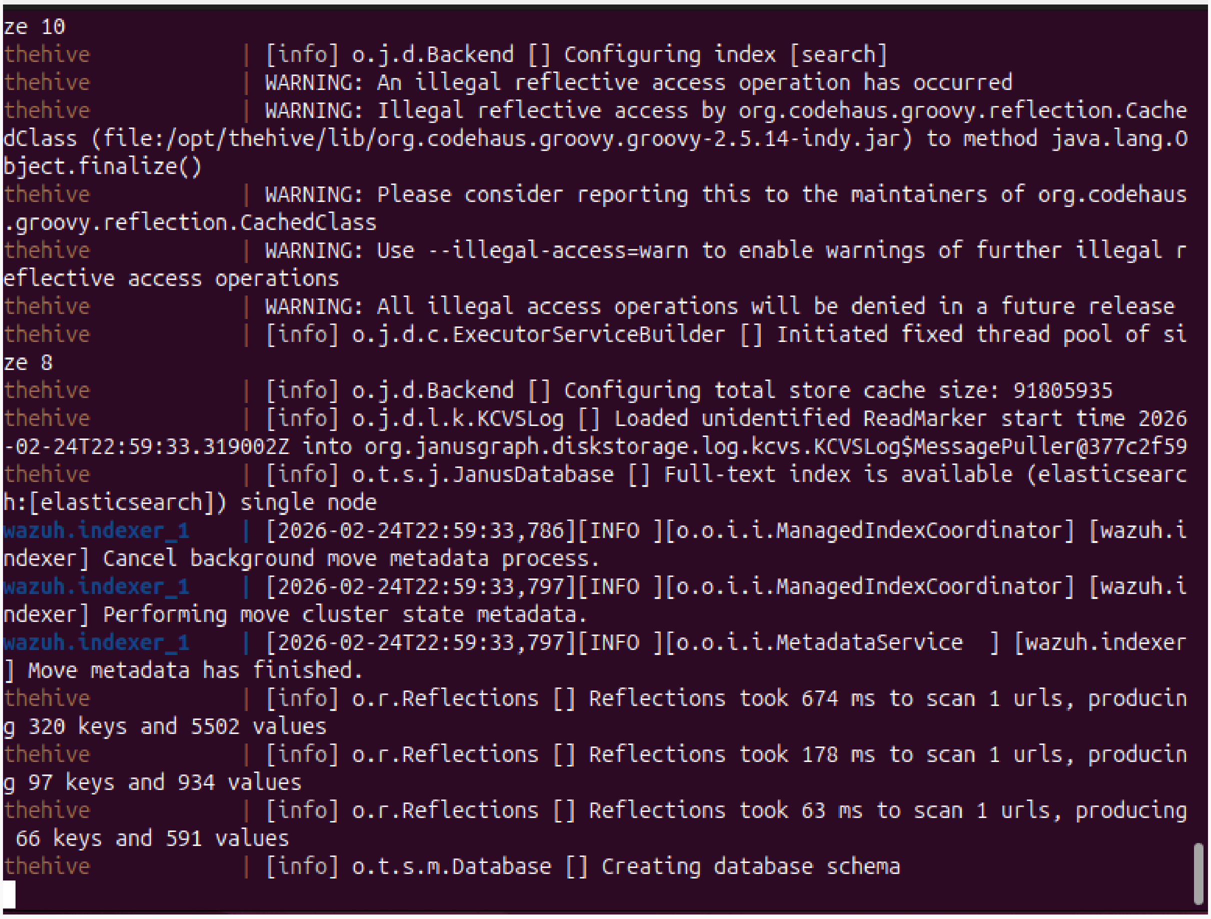
```
}

# ElasticSearch configuration
search {
 index = cortex
 # Name of the index
 uri = "http://elasticsearch:9200/"
}

# Cache configuration
cache.job = 10 minutes
cache.user = 5 minutes
cache.organization = 5 minutes

job {
 runner = [docker]
}

# Docker job runner configuration
dockerJobRunner {
 # Directory where job files are located
 directory = /tmp/cortex-jobs
 # Docker image timeout
 timeout = 30 minutes
 # Path to the docker executable
 dockerExecutable = "docker"
}

analyzer {
 config {
   # HTTP proxy configuration
   # proxy.host = proxy.example.com
   # proxy.port = 3128

   # HTTPS proxy configuration
   # proxy.https.host = proxy.example.com
   # proxy.https.port = 3128

   # Proxy authentication
   # proxy.auth.username = username
   # proxy.auth.password = password

   # Ignore proxy for these hosts
   # proxy.nonProxyHosts = ["localhost", "127.0.0.1"]
 }
 # Analyzer timeouts
 timeout = 120 seconds

 # Fork join pool for analyzers
 fork-join-executor {
   parallelism-factor = 2.0
   parallelism-max = 4
 }
}

# Authentication configuration
auth {
 provider = [local]

 # Multi-factor authentication
 multifactor = [totp]

 # Session timeout
 session.warning = 5m
 session.inactivity = 1h
}

# Datastore configuration
datastore {
 name = data
 # Size of stored files
 chunksize = 1m
 hash {
   main = "SHA-256"
   extra = ["SHA-1", "MD5"]
 }
 attachment.password = "malware"
}

# Maximum textual content length
play.http.parser.maxMemoryBuffer = 1M
play.http.parser.maxDiskBuffer = 1G
```

Este código añade los analizadores del servicio cortex.

Finalmente, lanzamos el comando para ejecutar y construir los contenedores con:

```
docker-compose up
```

Con ese comando tal como está, podrás visualizar todo el proceso de construcción, desde el pull de recursos y las descargas de imágenes de contenedores, hasta la configuración e incluso los logs en vivo de todos los servicios que se han configurado en el `docker-compose.yml`.

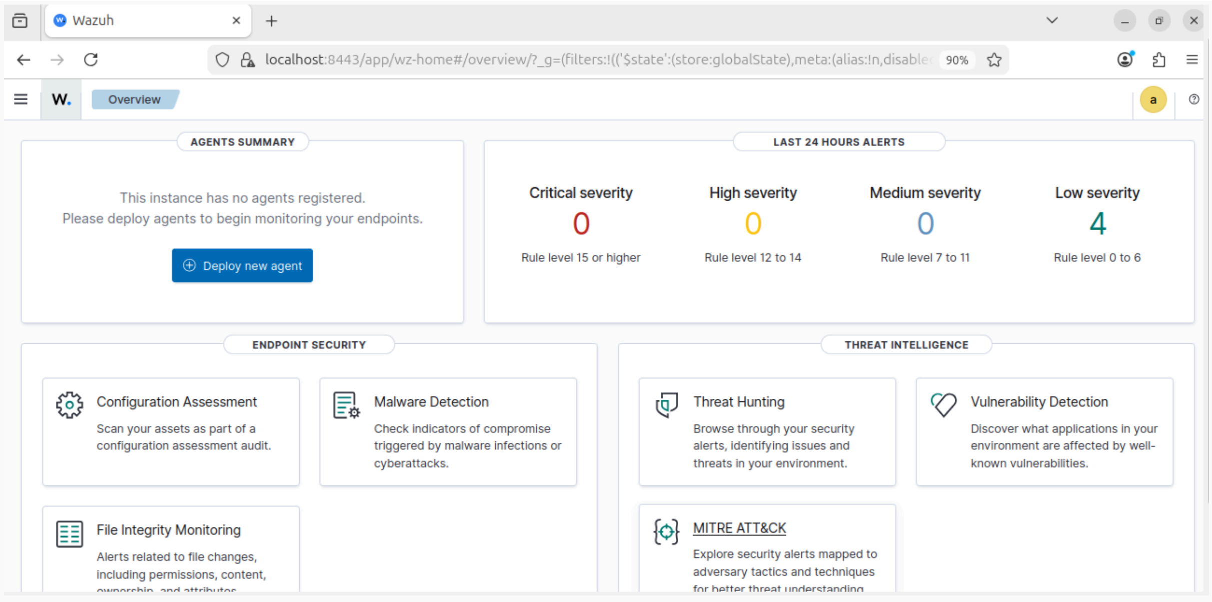


```
ze 10
thehive               | [info] o.j.d.Backend [] Configuring index [search]
thehive               | WARNING: An illegal reflective access operation has occurred
thehive               | WARNING: Illegal reflective access by org.codehaus.groovy.reflection.Cache
dClass (file:/opt/thehive/lib/org.codehaus.groovy.groovy-2.5.14-indy.jar) to method java.lang.O
bject.finalize()
thehive               | WARNING: Please consider reporting this to the maintainers of org.codehaus
.groovy.reflection.CachedClass
thehive               | WARNING: Use --illegal-access=warn to enable warnings of further illegal r
eflective access operations
thehive               | WARNING: All illegal access operations will be denied in a future release
thehive               | [info] o.j.d.c.ExecutorServiceBuilder [] Initiated fixed thread pool of si
ze 8
thehive               | [info] o.j.d.Backend [] Configuring total store cache size: 91805935
thehive               | [info] o.j.d.l.k.KCVSLog [] Loaded unidentified ReadMarker start time 2026
-02-24T22:59:33.319002Z into org.janusgraph.diskstorage.log.kcvs.KCVSLog$MessagePuller@377c2f59
thehive               | [info] o.t.s.j.JanusDatabase [] Full-text index is available (elasticsearc
h:[elasticsearch]) single node
wazuh.indexer_1       | [2026-02-24T22:59:33,786][INFO ][o.o.i.i.ManagedIndexCoordinator] [wazuh.i
ndexer] Cancel background move metadata process.
wazuh.indexer_1       | [2026-02-24T22:59:33,797][INFO ][o.o.i.i.ManagedIndexCoordinator] [wazuh.i
ndexer] Performing move cluster state metadata.
wazuh.indexer_1       | [2026-02-24T22:59:33,797][INFO ][o.o.i.i.MetadataService  ] [wazuh.indexer
] Move metadata has finished.
thehive               | [info] o.r.Reflections [] Reflections took 674 ms to scan 1 urls, producin
g 320 keys and 5502 values
thehive               | [info] o.r.Reflections [] Reflections took 178 ms to scan 1 urls, producin
g 97 keys and 934 values
thehive               | [info] o.r.Reflections [] Reflections took 63 ms to scan 1 urls, producing
 66 keys and 591 values
thehive               | [info] o.t.s.m.Database [] Creating database schema
```

Resultados

Accediendo desde el localhost de nuestra máquina y con los puertos configurados en el docker-compose.yml, podremos acceder a los servicios construidos.

localhost:8443 | Wahuh Dashboard



This instance has no agents registered.
Please deploy agents to begin monitoring your endpoints.

Critical severity	High severity	Medium severity	Low severity
0	0	0	4
Rule level 15 or higher	Rule level 12 to 14	Rule level 7 to 11	Rule level 0 to 6

Configuration Assessment
Scan your assets as part of a configuration assessment audit.

Malware Detection
Check indicators of compromise triggered by malware infections or cyberattacks.

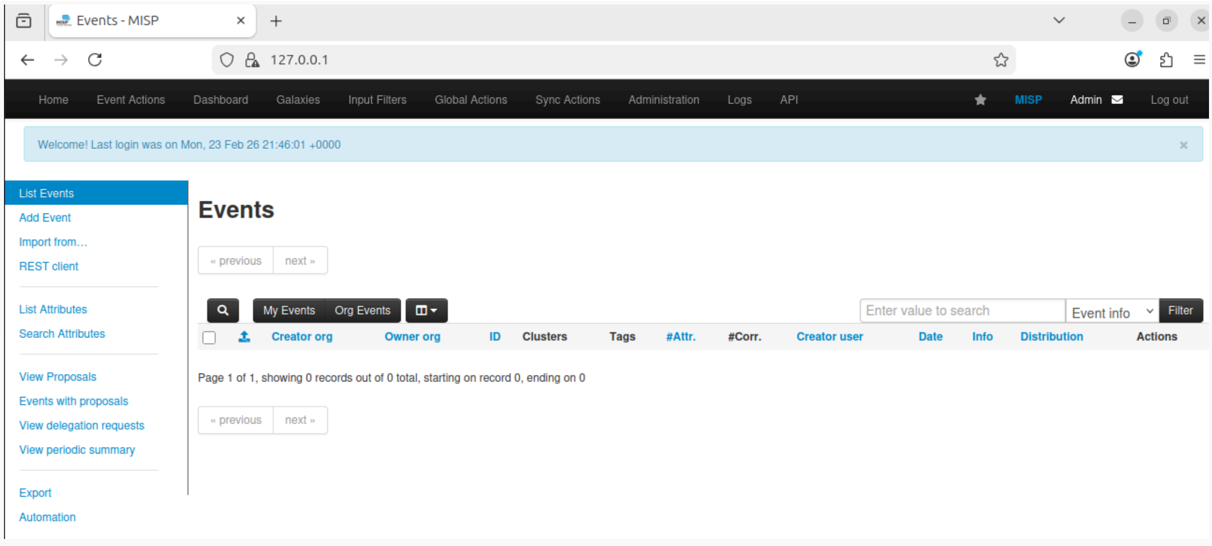
File Integrity Monitoring
Alerts related to file changes, including permissions, content, ownership, and attributes

Threat Hunting
Browse through your security alerts, identifying issues and threats in your environment.

Vulnerability Detection
Discover what applications in your environment are affected by well-known vulnerabilities.

MITRE ATT&CK
Explore security alerts mapped to adversary tactics and techniques for better threat understanding

localhost:443 | MISP

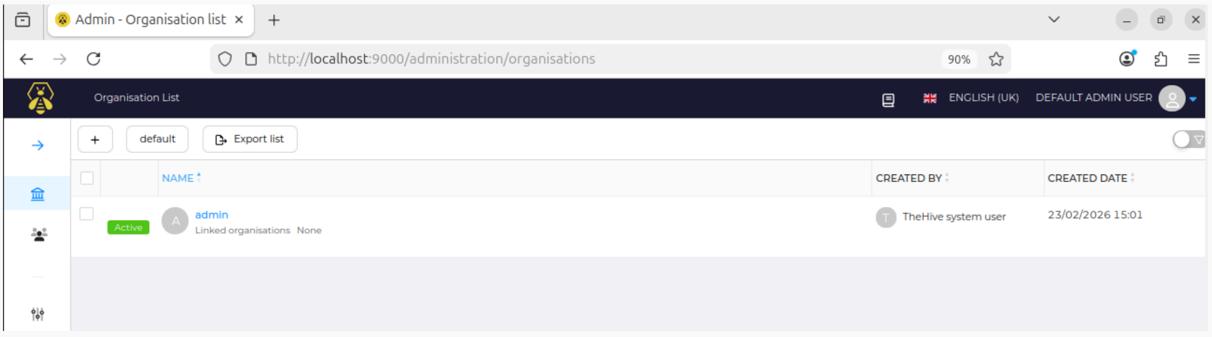


Welcome! Last login was on Mon, 23 Feb 26 21:46:01 +0000

Events

	Creator org	Owner org	ID	Clusters	Tags	#Attr.	#Corr.	Creator user	Date	Info	Distribution	Actions

Page 1 of 1, showing 0 records out of 0 total, starting on record 0, ending on 0 localhost:9000 | TheHive


Admin - Organisation list
http://localhost:9000/administration/organisations
90%
Organisation List
ENGLISH (UK)
DEFAULT ADMIN USER
+
default
Export list

	NAME	CREATED BY	CREATED DATE
Active	admin Linked organisations None	TheHive system user	23/02/2026 15:01