

Cristian Pérez Gómez

DevOps Empresarial: Despliegue de Sistema SOC automatizado

splunk® >

 Cortex


MISP

 TheHive

Índice

| | |
|--|------------------------------|
| Fase 1: Planificación y Preparación..... | Página 4 - Página 7 |
| Conocimientos básicos..... | Página 4 |
| Descripción de Herramientas..... | Página 5 - Página 6 |
| Descripción de Flujo de Trabajo..... | Página 7 |
| Fase 2: Despliegue de Herramientas..... | Página 8 - Página 24 |
| Despliegue de Splunk..... | Página 8 - Página 10 |
| Despliegue de TheHive..... | Página 11 - Página 15 |
| Despliegue de Cortex..... | Página 16 - Página 19 |
| Despliegue de Misp..... | Página 20 - Página 24 |
| Fase 3: Integraciones de Herramientas..... | Página 25 - Página 57 |
| Integracion TheHive + Cortex..... | Página 25 - Página 32 |
| Integracion Misp + (TheHive + Cortex)..... | Página 33 - Página 39 |
| Integración Splunk + (TheHive + Cortex)..... | Página 40 - Página 52 |
| Integracion Splunk + Misp..... | Página 53 - Página 57 |
| Fase 4: Prueba de integración en caso práctico..... | Página 58 - Página 87 |
| Creación de recolector HEC..... | Página 58 - Página 61 |
| Despliegue de entorno de pruebas..... | Página 62 - Página 63 |
| Creación de sistema de respuesta automático..... | Página 64 - Página 71 |
| Configuraciones específicas..... | Página 72 - Página 82 |
| Visualización de Workflow en funcionamiento..... | Página 83 - Página 85 |
| Conclusión y aclaraciones..... | Página 86 - Página 87 |
| Webgrafía..... | Página - 88 |

Fase 1: Planificación y Preparación

Conocimientos básicos

¿Qué es un sistema SOC?

Un **SOC (Security Operations Center)** es un centro de operaciones de seguridad donde un equipo especializado monitorea, detecta, analiza y responde a amenazas de ciberseguridad en tiempo real.

Su objetivo principal es proteger la infraestructura de TI de una organización mediante la vigilancia continua y la respuesta a incidentes.

Funciones principales de un SOC:

- **Monitoreo y Detección** → Usa herramientas como SIEM (Security Information and Event Management) para recolectar y analizar logs en busca de actividades sospechosas.
- **Análisis de Amenazas** → Investiga eventos de seguridad para determinar si representan un riesgo real.
- **Respuesta a Incidentes** → Actúa ante ataques o incidentes de seguridad para mitigarlos y contener su impacto.
- **Gestión de Vulnerabilidades** → Identifica y corrige fallas de seguridad en la infraestructura.
- **Automatización y Orquestación** → Usa herramientas como **TheHive, Cortex y MISP** (que ya tienes en Docker) para agilizar la respuesta a incidentes.
- **Inteligencia de Amenazas** → Recopila información sobre amenazas emergentes para mejorar la defensa del sistema.

¿Cuándo se usa este sistema?

El principal caso de uso de un SOC es la detección y respuesta a incidentes de ciberseguridad en tiempo real, protegiendo la infraestructura de TI contra ataques como ransomware, phishing, intrusiones y accesos no autorizados.

Descripción de Herramientas

Plataforma de desarrollo

Docker-compose es una herramienta que permite definir y administrar aplicaciones multicontenedor en Docker mediante un solo archivo YAML, facilitando la configuración y despliegue de servicios interdependientes.

Para un SOC, usar Docker Compose es ideal porque permite desplegar rápidamente herramientas como Splunk, TheHive, Cortex y MISP en contenedores aislados pero interconectados, garantizando escalabilidad, facilidad de mantenimiento y replicabilidad del entorno

Herramientas Principales

Splunk → Actúa como un “**SIEM**”, recopilando, indexando y analizando logs en tiempo real para detectar amenazas mediante reglas de correlación y análisis avanzado.

TheHive → Es una plataforma de gestión de incidentes, donde los analistas registran, investigan y coordinan la respuesta a eventos de seguridad.

Cortex → Es una plataforma de análisis y automatización, que permite ejecutar más de 100 analizadores “(como **VirusTotal**, **YARA**, etc)” sobre artefactos sospechosos y enriquecer la investigación de incidentes.

MISP → Funciona como una plataforma de inteligencia de amenazas “(**TIP**)”, centralizando información sobre indicadores de compromiso “(**IOCs**)” y compartiendo inteligencia con otros SOC's o equipos de respuesta a incidentes.

Herramientas de Almacenamiento

Mysql → Funciona como base de datos utilizada por MISP para almacenar datos.

Cassandra → Funciona como base de datos NoSQL utilizada por TheHive y Cortex para almacenar datos.

Minio → Funciona como un servidor de almacenamiento compatible con S3 utilizado por TheHive para almacenar archivos.

Herramientas de Búsqueda

ElasticSearch → Funciona como motor de búsqueda y análisis utilizado por TheHive para indexar y buscar datos.

Redis → Funciona como base de datos en memoria utilizada por MISP para mejorar el rendimiento.

Herramientas Adicionales

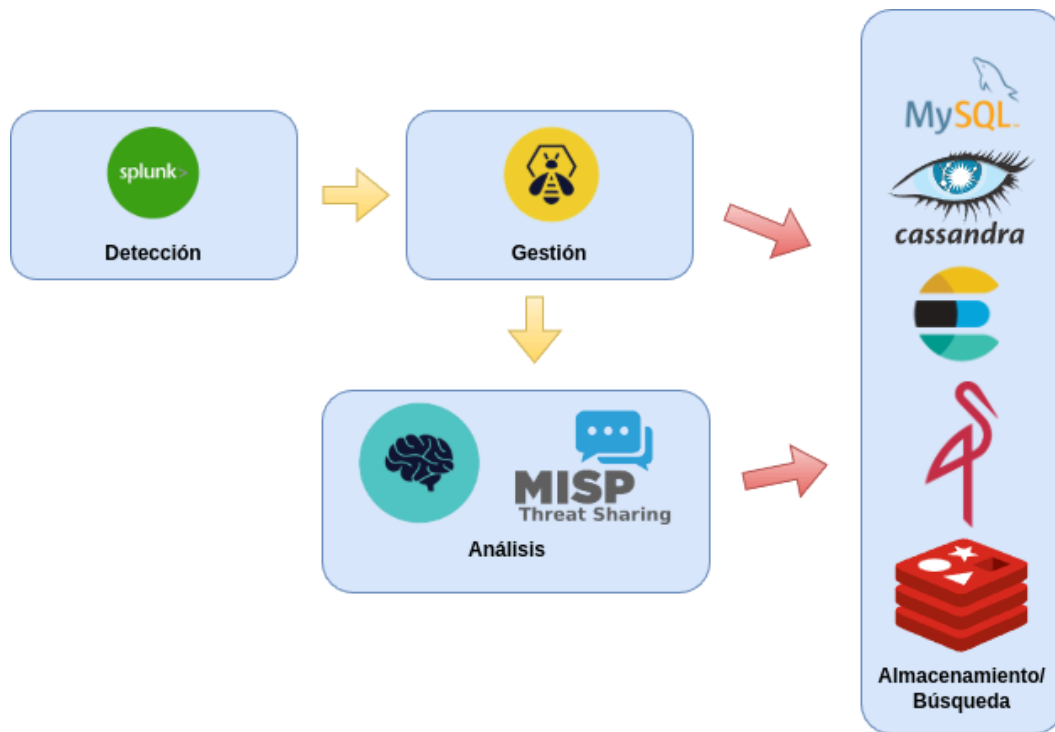
Misp-Modules → Son módulos adicionales para MISP que proporcionan capacidades de análisis y enriquecimiento.

TheHive - Cortex Plugin → Es un plugin disponible para Splunk utilizado para integrar la herramienta con instancias TheHive y Cortex.

Misp42 → Es un plugin disponible para Splunk utilizado para integrar la herramienta con instancias Misp.

Descripción de Flujo de Trabajo

Para desarrollar un sistema SOC, primero hay que tener claro el flujo de trabajo que debe tener el mismo. Comenzado por la función que cada herramienta desempeñará en él, las funciones se dividen principalmente en 5 Grupos.



Detección → Consiste en identificar posibles amenazas en la infraestructura de TI

Gestión → Una vez detectado un posible incidente, es necesario administrarlo de manera eficiente mediante plataformas de gestión de incidentes que permite organizar, priorizar y asignar casos al equipo de analistas de seguridad.

Análisis → El análisis de amenazas es crucial para entender el impacto y alcance del incidente. Para ello, se utilizan herramientas que ejecutan análisis automatizados.

Almacenamiento → Un SOC debe mantener un historial de eventos y amenazas para auditoría, cumplimiento normativo y aprendizaje continuo.

Búsqueda → Para facilitar la investigación y respuesta rápida, se necesitan herramientas que permitan realizar consultas avanzadas sobre eventos pasados y actuales.

Fase 2: Despliegue de Herramientas

Despliegue de Splunk

¿Qué es Splunk?

Splunk es una plataforma de software utilizada para la recopilación, análisis y visualización de datos generados por máquinas en tiempo real. Se usa principalmente para monitoreo de seguridad, análisis de registros (logs), inteligencia operativa y análisis de grandes volúmenes de datos no estructurados.

Splunk permite indexar, buscar y correlacionar datos de diversas fuentes, como servidores, aplicaciones, dispositivos de red y sensores IoT, proporcionando información valiosa para la toma de decisiones y la seguridad informática.

Características principales de Splunk

- **Ingesta de datos en tiempo real** → Capta y procesa grandes volúmenes de datos estructurados y no estructurados.
- **Búsqueda y análisis potente** → Usa un lenguaje de búsqueda “(SPL - Splunk Processing Language)” para filtrar, correlacionar y extraer información de los logs.
- **Visualización de datos** → Genera dashboards interactivos, gráficos y reportes personalizados.
- **Alertas y automatización** → Permite configurar alertas basadas en eventos específicos y ejecutar acciones automatizadas.
- **Escalabilidad** → Puede manejar grandes volúmenes de datos, ideal para entornos empresariales y de alta disponibilidad.

Splunk será la herramienta principal en nuestro proyecto, ya que a través de él se visualizarán todos los datos y en base a ellos se tomarán decisiones en base a las reglas que establezcamos, en nuestro caso, levantaremos este servicio en un contenedor, para ello requerimos de la configuración mostrada en la siguiente imagen.

```
splunk:
  image: splunk/splunk:latest
  restart: unless-stopped
  environment:
    - SPLUNK_START_ARGS=--accept-license
    - SPLUNK_PASSWORD=
  volumes:
    - splunkdata:/opt/splunk/var
    - splunketc:/opt/splunk/etc
  ports:
    - "0.0.0.0:8000:8000"
    - "0.0.0.0:8088:8088"
  networks:
    - SOC_NET
```

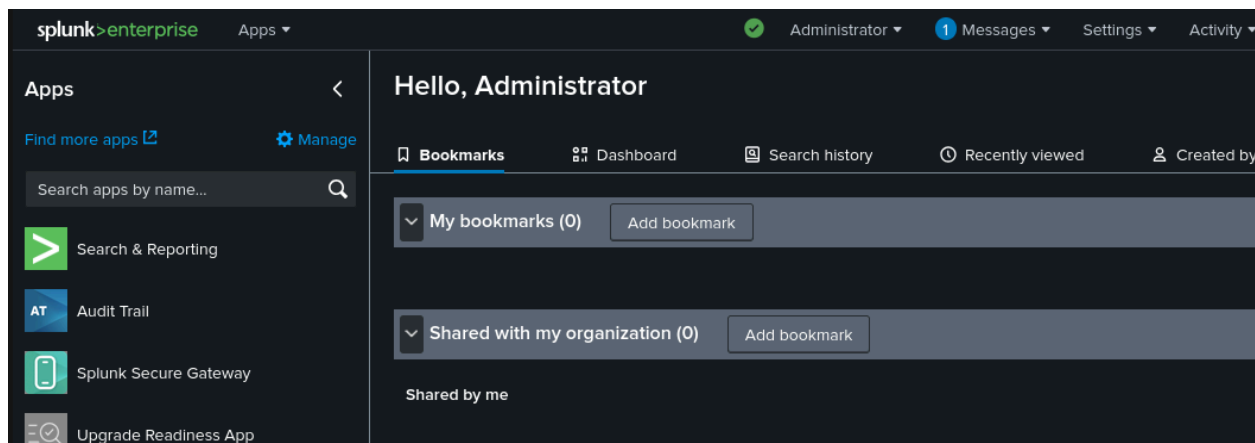
Algunos aspectos a considerar sobre la configuración que hemos observado son los siguientes.

- **Environment** → Esta sección establece las variables de entorno necesarias para configurar Splunk.
 - **SPLUNK_PASSWORD=your_password** → Define la contraseña de administración de Splunk. Se debe cambiar por una contraseña segura antes de usar el contenedor.
 - **SPLUNK_START_ARGS=--accept-license** → Acepta los términos de la licencia al iniciar el contenedor.
- **Ports** → Redirige puertos del contenedor a la máquina host.
 - **"8000:8000"** → El puerto 8000 del contenedor, que se usa para la interfaz web de Splunk, se mapea al puerto 8000 de la máquina host.
 - **"8088:8088"** → El puerto 8088, que es utilizado por el HTTP Event Collector (para recibir logs), se mapea al mismo puerto en el host.

Una vez hemos configurado el archivo “.yml” necesario para splunk, lo único que quedará es acceder al puerto de administración que hemos expuesto para acceder al panel web, donde tendremos que introducir las credenciales indicadas en el archivo previamente mencionado para iniciar sesión por primera vez.



Una vez nos hayamos autenticado en Splunk, ya tendremos acceso a su panel de administración, donde podremos observar las principales aplicaciones que tenemos disponibles. De estas hablaremos posteriormente con mucha más profundidad.



Despliegue de TheHive

¿Qué es TheHive?

TheHive es una plataforma de código abierto diseñada para la gestión de incidentes de seguridad y la orquestación de respuestas.

Está orientada a la gestión de incidentes de seguridad informática y es muy utilizada en entornos de SOC “**(Security Operations Centers)**” para la recopilación de datos, el análisis y la respuesta a eventos de seguridad en tiempo real.

TheHive permite a los equipos de seguridad gestionar casos de incidentes, colaborar de manera eficiente, e integrar diferentes herramientas de seguridad para mejorar la efectividad de la respuesta. Además, se integra con plataformas de SIEM, como Splunk y ELK, para centralizar y correlacionar los datos de seguridad.

Características principales de TheHive

- **Gestión de Incidentes** → Permite crear, gestionar y hacer seguimiento de los casos de incidentes de seguridad. Los usuarios pueden asociar eventos de seguridad con casos y agregar información adicional, como comentarios y evidencias.
- **Automatización de Tareas** → TheHive puede automatizar ciertas tareas mediante plantillas o integraciones con otras herramientas. Se pueden definir flujos de trabajo personalizados para la respuesta ante incidentes.
- **Integración con Otras Herramientas de Seguridad** → Se integra con herramientas como MISP (“**Malware Information Sharing Platform**”), Cortex “**(para automatizar la respuesta y análisis)**”, SIEMs y herramientas de análisis forense. Esto permite ampliar su funcionalidad y proporcionar

Como hemos mencionado anteriormente, utilizaremos TheHive para gestión de incidentes de seguridad y la orquestación de respuestas a eventos de seguridad en tiempo real.

```
services:
  thehive:
    image: strangebee/thehive:5.2
    restart: unless-stopped
    depends_on:
      - cassandra
      - elasticsearch
      - minio
      - cortex.local
    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:
      - thehivedata:/etc/thehive/application.conf
      - thehive logs:/var/log/thehive
      - thehivefiles:/opt/thehive/files
    networks:
      - SOC_NET
```

Es necesario aclarar que TheHive necesita de un contenedor Elasticsearch que utiliza para indexar y buscar datos.

```
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
    - elasticsearchlogs:/var/log/elasticsearch
  networks:
    - SOC_NET
```

También es necesario aclarar que TheHive necesita de un contenedor Elasticsearch que utiliza para almacenar datos.

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

También es necesario aclarar que TheHive necesita de un contenedor Minio que utiliza para almacenar datos.

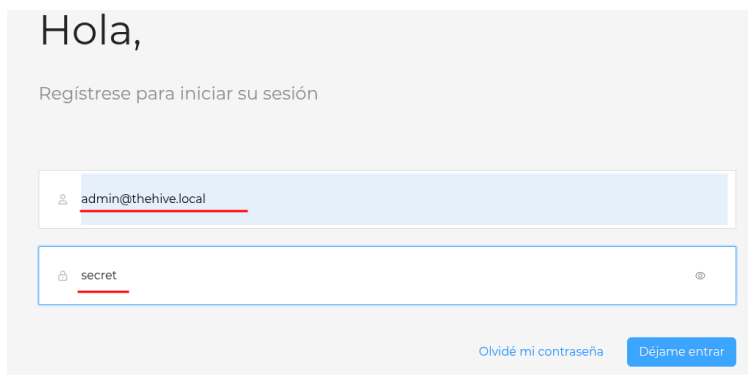
```
minio:
  image: quay.io/minio/minio
  restart: unless-stopped
  command: ["minio", "server", "/data", "--console-address", ":9002"]
  environment:
    - MINIO_ROOT_USER=minioadmin
    - MINIO_ROOT_PASSWORD=minioadmin
  ports:
    - "0.0.0.0:9002:9002"
  volumes:
    - "miniodata:/data"
    - minioconfig:/root/.minio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

En cuanto despleguemos los servicios al acceder al puerto de administración que hemos expuesto para acceder al panel web, tendremos que introducir el usuario y contraseña por defecto para poder acceder al propio panel web.

Las credenciales para acceder por primera vez son:

- **Usuario** → admin@thehive.local
- **Contraseña** → secret

No obstante, se recomienda encarecidamente modificar las credenciales por defecto.



Hola,

Regístrese para iniciar su sesión

[Olvidé mi contraseña](#) [Déjame entrar](#)

Posteriormente ya podremos acceder al servicio ingresando las credenciales que hemos creado anteriormente.

Una vez hemos ingresado las credenciales que hemos creado anteriormente, ya tendremos acceso al panel de administración de TheHive, en este entraremos mucho más a profundidad posteriormente, pero ya podremos empezar a ver las múltiples funcionalidades de la herramienta.

The screenshot displays the 'Lista de organizaciones' (List of organizations) page in TheHive. The interface features a dark blue header with the TheHive logo and the page title. Below the header, there is a sidebar with navigation icons, including a blue arrow, a building icon, a group of people icon, a gear icon, and a wrench icon. The main content area contains a table with the following structure:

| <input type="checkbox"/> | | NOMBRE ↕ |
|--------------------------|--------|---|
| <input type="checkbox"/> | Activo | <div><div>A</div><div>admin</div><div>Organizaciones vinculadas Ninguno</div></div> |

Despliegue de Cortex

¿Qué es Cortex?

Cortex es una plataforma de código abierto diseñada para la automatización de la respuesta ante incidentes de seguridad.

Está integrada principalmente con TheHive, formando una solución completa para la gestión de incidentes y la orquestación de respuestas automatizadas.

Cortex permite ejecutar análisis avanzados sobre los incidentes detectados, buscar información sobre amenazas y ejecutar respuestas de seguridad automáticas, todo dentro de un marco controlado.

Su uso principal es en Security Operations Centers “(SOCs)” y en organizaciones que buscan automatizar la remediación de incidentes, permitiendo una respuesta más rápida y eficaz a las amenazas de seguridad.

Características principales de Cortex

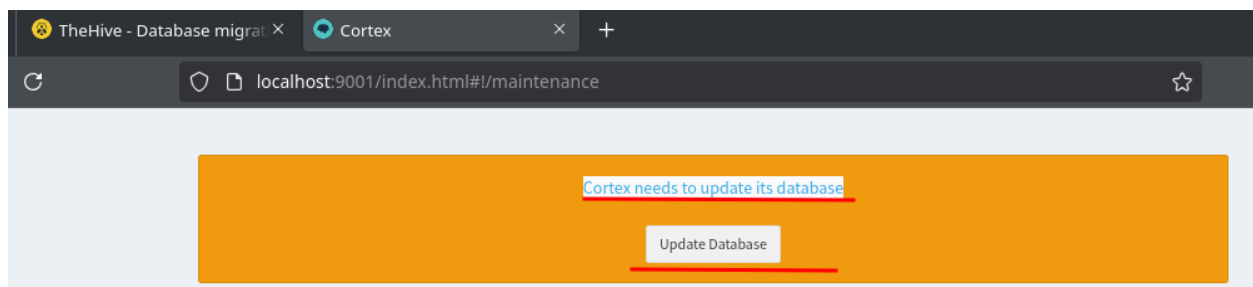
- **Automatización de Respuestas a Incidentes** → Cortex permite la automatización de tareas y respuestas ante incidentes de seguridad. Por ejemplo, puede ejecutar scripts, bloquear direcciones IP, o realizar consultas a bases de datos de inteligencia de amenazas para obtener información relacionada con los incidentes.
- **Integración con TheHive** → Cortex está diseñado para integrarse perfectamente con TheHive, permitiendo que las respuestas automatizadas se ejecuten directamente dentro de los casos de TheHive. Esto crea un flujo de trabajo coherente donde la información y las acciones se gestionan en un único entorno.

Cortex será el encargado de orquestar las respuestas automatizadas, para su despliegue inicial deberemos contar con él en nuestro archivo ".yml", un ejemplo de este sería la configuración que podemos apreciar a continuación.

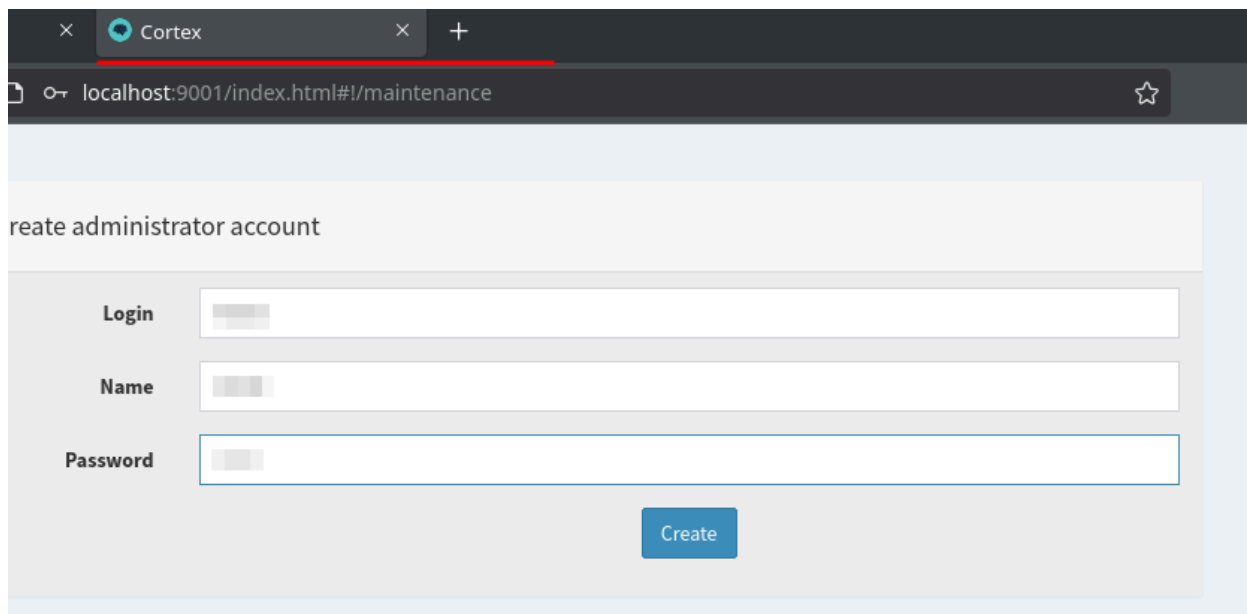
```
cortex.local:
  image: thehiveproject/cortex:latest
  restart: unless-stopped
  environment:
    - job_directory=/tmp/cortex-jobs
    - docker_job_directory=/tmp/cortex-jobs
  volumes:
    - /var/run/docker.sock:/var/run/docker.sock
    - /tmp/cortex-jobs:/tmp/cortex-jobs
    - ./cortex/logs:/var/log/cortex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

En cuanto desplaguemos los servicios al acceder al puerto de administración que hemos expuesto para acceder al panel web, tendremos que actualizar la base de datos para continuar.

Es necesario aclarar que Cortex también necesita de un contenedor Elasticsearch que utilizan como backend para almacenar y consultar los datos relacionados con los incidentes de seguridad y otros eventos, por lo tanto, será necesario incluir a este contenedor en nuestro archivo también.

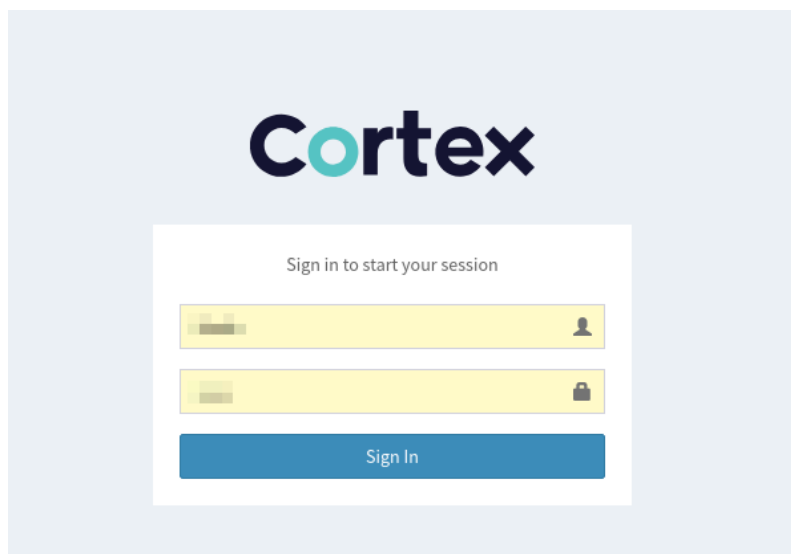


Posteriormente, tendremos que crear las credenciales de acceso de administrador para poder acceder al panel web, para ello, deberemos de rellenar los campos que se observan a continuación en la imagen.



The screenshot shows a web browser window with the title 'Cortex'. The address bar displays 'localhost:9001/index.html#/maintenance'. The main content area is titled 'create administrator account' and contains three input fields: 'Login', 'Name', and 'Password'. Each field has a small grey icon to its left. Below the fields is a blue 'Create' button.

Posteriormente ya podremos acceder al servicio ingresando las credenciales que hemos creado anteriormente.



The screenshot shows the Cortex logo at the top. Below it is a white box with the text 'Sign in to start your session'. Inside this box are two input fields: the first has a person icon and the second has a lock icon. Below these fields is a blue 'Sign In' button.

Despliegue de Misp

¿Qué es Misp?

MISP “(**Malware Information Sharing Platform & Threat Sharing**)” es una plataforma de código abierto diseñada para el intercambio de información sobre amenazas cibernéticas “(**CTI, Cyber Threat Intelligence**)”.

Permite a organizaciones, equipos de seguridad y gobiernos compartir, almacenar y analizar indicadores de compromiso (IoCs) y otros datos de inteligencia de amenazas.

Características principales de Misp

- **Intercambio de inteligencia de amenazas** → Facilita la colaboración entre organizaciones al compartir información sobre amenazas, malware, ataques dirigidos y otros incidentes de seguridad.
- **Estructura de datos flexible y enriquecida** → Usa un modelo de datos estructurado con eventos y atributos para representar información de amenazas de manera clara y reutilizable.
- **Automatización y API REST** → Dispone de una API REST completa para la integración con otros sistemas de seguridad como SIEMs “(**Splunk, ELK**)”, SOARs “(**TheHive, Cortex**)”, IDS/IPS y herramientas de análisis forense.
- **Colaboración y control de acceso** → Ofrece mecanismos de control de acceso para definir quién puede ver, modificar o compartir información dentro de una comunidad.
- **Extensibilidad mediante módulos** → Integra módulos de expansión y enriquecimiento que permiten consultar fuentes externas como VirusTotal, Shodan, WHOIS, entre otros.

Para desplegar Misp requerimos de incluir el siguiente código a nuestro archivo ".yml", este será expuesto por el puerto 80 de nuestro equipo.

```
misp.local:
  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"
    - mispdata:/var/www/MISP/data
  environment:
    - MYSQL_HOST=misp_mysql
    - MYSQL_DATABASE=mispdb
    - MYSQL_USER=
    - MYSQL_PASSWORD=
    - MISP_ADMIN_EMAIL=
    - MISP_ADMIN_PASSPHRASE=
    - MISP_BASEURL=localhost
    - TIMEZONE=Europe/London
    - "INIT=true"
    - "CRON_USER_ID=1"
    - "REDIS_FQDN=redis"
    - "HOSTNAME=https://192.168.1.49" # CAMBIAR POR LA IP DE LA MÁQUINA
  networks:
    - SOC_NET
```

Es importante aclarar que la variable de entorno **"HOSTNAME"** tiene que ser sustituida por la dirección IP real del equipo donde estemos dockerizando los servicios, de lo contrario, no será expuesto correctamente.

También es necesario incluir las credenciales del contenedor Mysql, posteriormente explicaremos para qué utiliza Misp este contenedor.

Es necesario aclarar que Misp necesita de un contenedor MySQL que utiliza para almacenar datos.

```
misp_mysql:
  image: mysql/mysql-server:5.7
  restart: unless-stopped
  volumes:
    - mispsqldata:/var/lib/mysql
    - mispmysqllogs:/var/log/mysql
  environment:
    - MYSQL_DATABASE=
    - MYSQL_USER=
    - MYSQL_PASSWORD=
    - MYSQL_ROOT_PASSWORD=
  networks:
    - SOC_NET
```

También es necesario aclarar que Misp necesita de un contenedor Redis que se utiliza para mejorar el rendimiento en general.


```
redis:
  image: redis:latest
  networks:
    - SOC_NET
  volumes:
    - redisdata:/data
    - redislogs:/var/log/redis
```

También es necesario aclarar que Misp necesita de un contenedor de módulos adicionales que proporcionan capacidades de análisis y enriquecimiento.

```
misp-modules:
  image: coolacid/misp-docker:modules-latest
  environment:
    - "REDIS_BACKEND=redis"
  depends_on:
    - redis
    - misp_mysql
  networks:
    - SOC_NET
  volumes:
    - mispmoduledata:/var/www/MISP/modules/data
```

El siguiente paso será acceder al servicio y realizar el primer login en la plataforma, para ello, utilizaremos las credenciales indicadas en el archivo ".yaml".

Initial Install, please configure



MISP
Threat Sharing

Login

Email Password

Login

Posteriormente deberemos de modificar la contraseña por razones de seguridad ya que Misp es muy exigente en cuanto a requisitos de complejidad de contraseña.

Change Password

New password ⓘ Confirm new password

Confirm with your current password

Submit

Una vez cambiamos la contraseña, ya tendremos acceso a la aplicación, en este entraremos mucho más a profundidad posteriormente, pero ya podremos empezar a ver las múltiples funcionalidades de la herramienta.

The screenshot displays a web application interface. At the top, a dark navigation bar contains links: Home, Event Actions, Dashboard, Galaxies, Input Filters, Global Actions, and Sync Action. Below this, a green notification banner with a red underline states "Password Changed.". On the left, a sidebar menu lists options: Edit My Profile, Change Password, My Profile (highlighted in blue), My Settings, Periodic summary settings, Set Setting, List Organisations, Role Permissions, List Sharing Groups, Add Sharing Group, List Sharing Group Blueprints, and Add Sharing Group Blueprint. The main content area is titled "User admin@admin.test" and contains a table of user details:

| | | | | | | | | | |
|------------------------------|---|------------------------------|-----------------|---------------------|-----------------|----------------------|-----------------|-----------------------|-----------------|
| ID | 1 | | | | | | | | |
| Email | admin@admin.test | | | | | | | | |
| Organisation | ORGNAME | | | | | | | | |
| Role | admin | | | | | | | | |
| TOTP | No Generate | | | | | | | | |
| Email notifications | <table><tr><td>Event published notification</td><td>No</td></tr><tr><td>Daily notifications</td><td>No</td></tr><tr><td>Weekly notifications</td><td>No</td></tr><tr><td>Monthly notifications</td><td>No</td></tr></table> | Event published notification | No | Daily notifications | No | Weekly notifications | No | Monthly notifications | No |
| Event published notification | No | | | | | | | | |
| Daily notifications | No | | | | | | | | |
| Weekly notifications | No | | | | | | | | |
| Monthly notifications | No | | | | | | | | |
| Contact alert enabled | No | | | | | | | | |

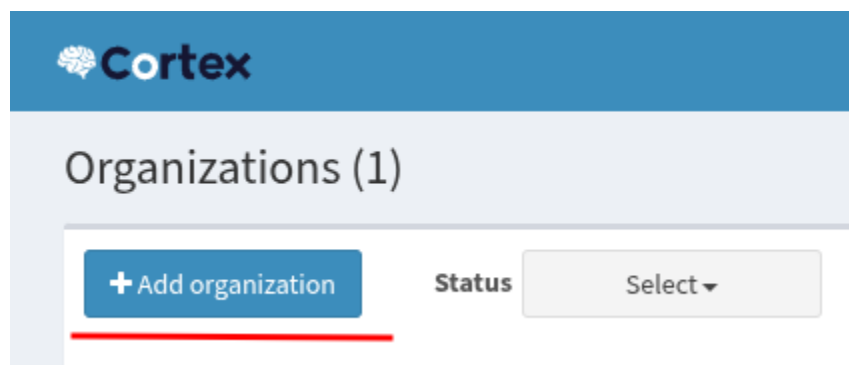
Debemos tener en cuenta que a partir de este momento deberemos iniciar sesión con el email mostrado en la anterior imagen "(a no ser que se haya modificado manualmente)" y la contraseña nueva que nosotros mismos hemos modificado.

Fase 3: Integraciones de Herramientas

Integración TheHive + Cortex

Llegados a este punto, ya tendremos todas las herramientas desplegadas, pero ninguna de ellas trabaja en conjunto, para lograr este objetivo debemos realizar ciertas configuraciones manuales.

Empezaremos por integrar TheHive con Cortex, para ello, en la interfaz de Cortex, pulsaremos en añadir una nueva organización.



Posteriormente, agregaremos un nombre y una descripción a esta nueva organización que estamos creando.

Create organization

Name *

TheHiveProject

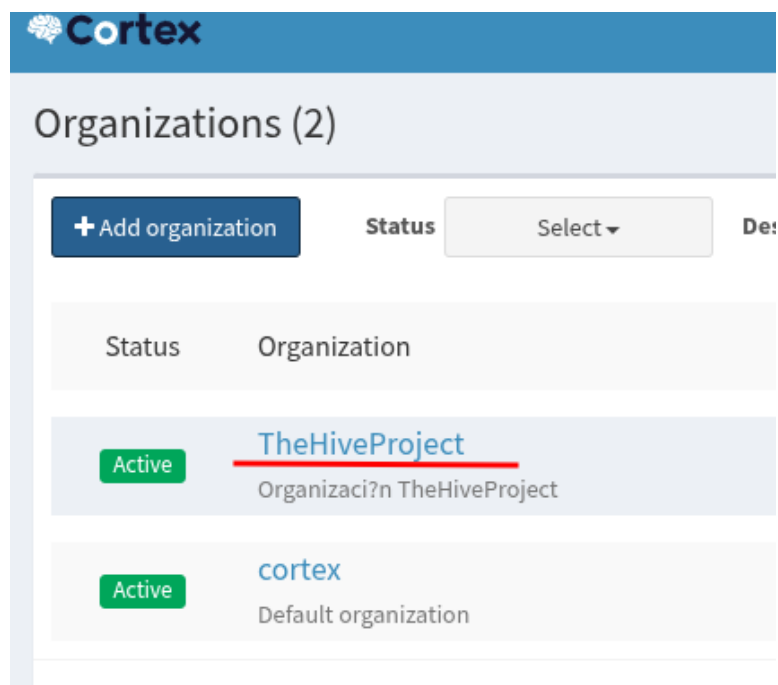
Description *

Organización TheHiveProject

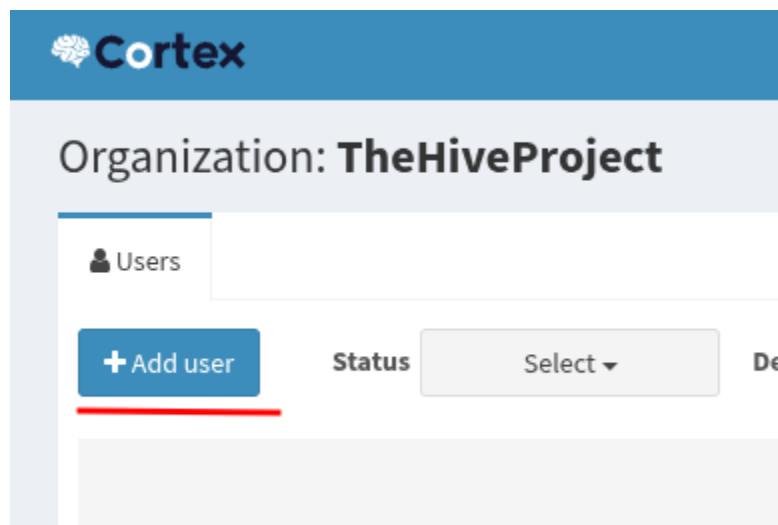
Cancel

* Required field

Una vez realizados los pasos anteriores, veremos que se ha creado la nueva organización con éxito.



El siguiente paso es crear un nuevo usuario para esta organización, para ello, pulsaremos en añadir un nuevo usuario.



Posteriormente le agregaremos a este nuevo usuario un nombre, un logueo y le asignaremos el rol que tendrá.

Los roles disponibles son:

- **ReadOnly** → Solo permite la lectura de datos.
- **Analyze** → Permite la lectura de datos y el análisis de los mismos.
- **OrgAdmin** → Dentro de su organización, tiene todos los permisos.

En nuestro caso elegiremos “**OrgAdmin**”, ya que las operaciones que vamos a realizar requieren de todos los permisos.

Add user

Login *

TheHive_Cortex_Connector@local.com

Full name *

TheHive_Cortex_Connector

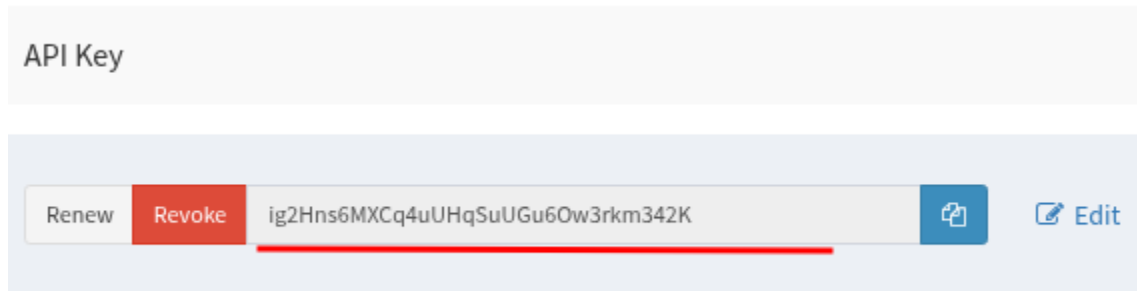
Roles *

read, analyze, orgadmin ▼

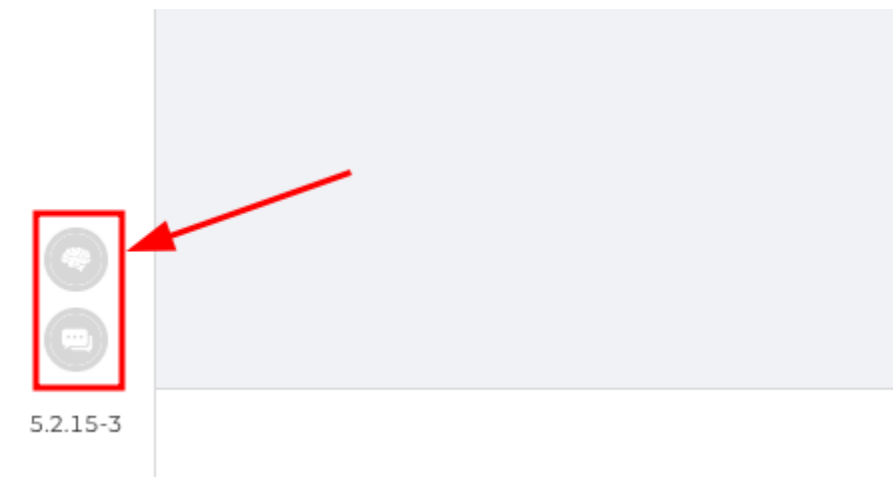
Para lograr unir los servicios, usaremos “**API Keys**”, que son tokens de verificación que usan los servicios para autenticarse entre sí. Al nuevo usuario creado le crearemos una API Key de la siguiente forma:

| Password | API Key |
|-------------------------|---------------------------|
| <div>New password</div> | <div>Create API Key</div> |

Una vez pulsado en botón **“Create API Key”** se nos creará la misma, es importante copiarla ya que la usaremos posteriormente.



Ahora pasaremos al panel de control de TheHive, si nos fijamos bien en la parte inferior izquierda del panel veremos estos dos iconos apagados, esto significa que de momento, no tiene una conexión funcional con Cortex o con Misp.





También podemos comprobar en el apartado de licencia que no tenemos servidores unidos con el servicio.


Adicionalmente podremos comprobar en este apartado otros valores como usuarios, organizaciones, plantillas etc.

Además, podremos agregar una licencia de producto si es que la tenemos, no es nuestro caso, por eso utilizaremos la versión community, que aunque es limitada en cuanto a espacio, no está ilimitada en cuanto a funciones.

Gestión de plataforma


 Licencia

 Estado

 Marca

Corteza

MISP

 Autenticación

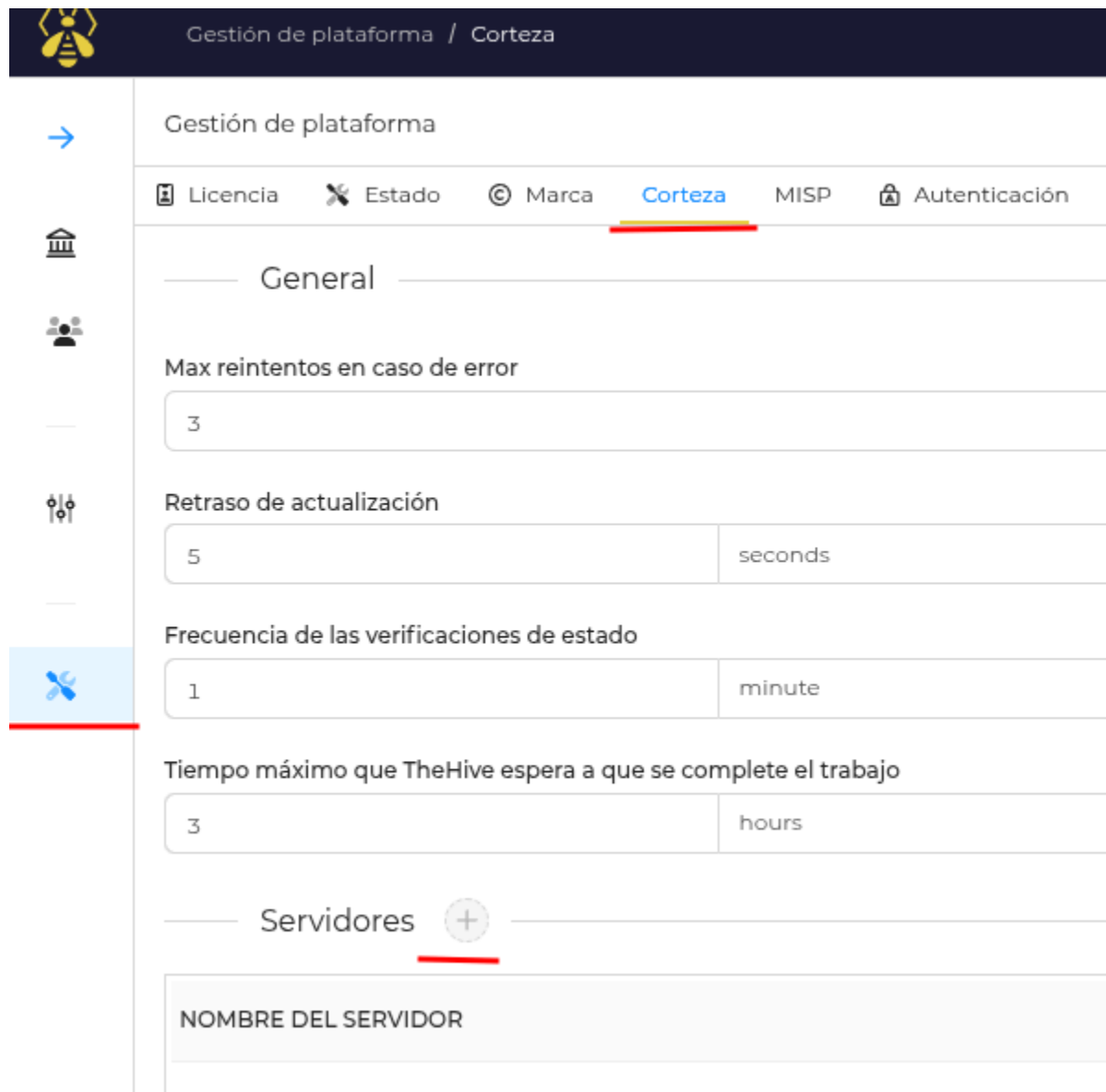
Gestión de licencias

Sin licencia

Activar una licencia

| | |
|--------------------------|---------------|
| Usuario | community |
| Usuarios de solo lectura | 0 / Ilimitado |
| Usuarios normales | 0 / 2 |
| Usuarios del servicio | 0 / Ilimitado |
| Organizaciones | 0 / 1 |
| Cuadros de mando | 0 / Ilimitado |
| Plantillas de casos | 0 / Ilimitado |
| Cluster de nodos | 1 / 1 |
| Servidores MISP | 0 / 1 |
| Servidores Cortex | 0 / 1 |

Para integrar el servicio cortex con TheHive, nos dirigimos al apartado llamado con su propio nombre en el apartado de configuraciones del servidor en TheHive.



The screenshot shows the 'Gestión de plataforma / Corteza' configuration page in TheHive. The left sidebar contains navigation icons: a blue arrow, a building, a group of people, a gear, and a wrench. The main content area has a top navigation bar with 'Licencia', 'Estado', 'Marca', 'Corteza' (highlighted with a red underline), 'MISP', and 'Autenticación'. Below this is a 'General' section with the following settings:

- Max reintentos en caso de error:** A text input field containing the value '3'.
- Retraso de actualización:** A text input field containing '5' and a dropdown menu set to 'seconds'.
- Frecuencia de las verificaciones de estado:** A text input field containing '1' and a dropdown menu set to 'minute'.
- Tiempo máximo que TheHive espera a que se complete el trabajo:** A text input field containing '3' and a dropdown menu set to 'hours'.

Below the 'General' section is a 'Servidores' section, indicated by a red underline and a '+' icon. It contains a single text input field with the placeholder text 'NOMBRE DEL SERVIDOR'.

Aquí también podremos modificar diferentes valores generales de la conexión con Cortex en caso de que sea necesario.

Una vez pulsado en el botón “+”, añadiremos el nombre que va a tener el servidor en la interfaz, la URL completa del servicio de Cortex “(**Con la IP del contenedor**)” y la clave API que hemos copiado anteriormente.

Configura el servidor "{nombre}"

General

Nombre del servidor

cortex0

* URL del servidor

http://172.18.0.10:9001

* Clave API

.....

Posteriormente, ejecutaremos una prueba de conexión con el otro contenedor Cortex.

Ajustes avanzados

Elija el filtro en las organizaciones de TheHive

Incluir todas las organizaciones

Cancelar

Prueba de conexión al servidor

Agregar

La prueba debe devolver este resultado, de lo contrario será necesario verificar la configuración de la conexión o de los servicios.

success



La configuración de Cortex se ha probado con éxito

Una vez realizados estos pasos, ya tendremos integrados ambos servicios, lo podemos comprobar en el apartado de licencia y el símbolo del cerebro se debe poner en color verde.

Gestión de plataforma

[Licencia](#) Estado Marca Corteza MISP

Gestión de licencias

Sin licencia

Activar una licencia

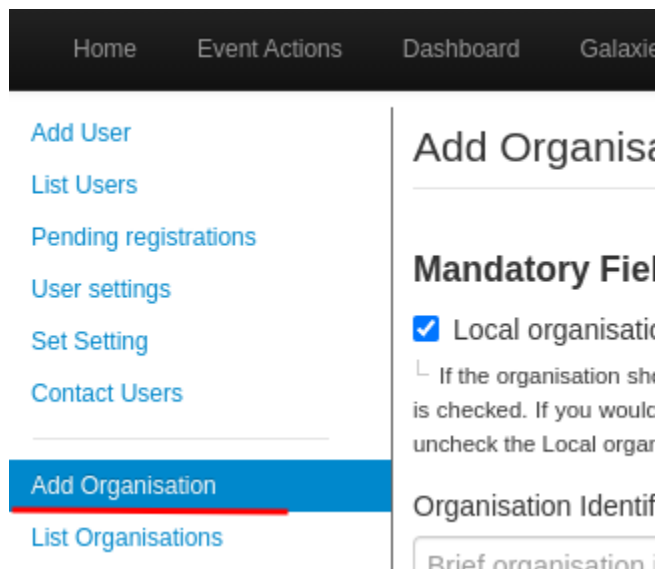
| | |
|--------------------------|---------------|
| Usuario | community |
| Usuarios de solo lectura | 0 / Ilimitado |
| Usuarios normales | 0 / 2 |
| Usuarios del servicio | 0 / Ilimitado |
| Organizaciones | 0 / 1 |
| Cuadros de mando | 0 / Ilimitado |
| Plantillas de casos | 0 / Ilimitado |
| Cluster de nodos | 1 / 1 |
| Servidores MISP | 0 / 1 |
| Servidores Cortex | 1 / 1 |



5.2.15-3

Integracion Misp + (TheHive + Cortex)

Una vez realizada la integración anterior, pasaremos a integrar Misp a ambas herramientas para que trabajen en conjunto, para ello, en la interfaz de Misp, pulsaremos en añadir una nueva organización.



Posteriormente asignaremos un nombre a la nueva organización y generaremos un “UUID” de forma automática a la misma.

A screenshot of the 'Add Organisation' form in Misp. The form is titled 'Add Organisation' and has a 'Mandatory Fields' section. In this section, the 'Local organisation' checkbox is checked. Below it, there is a text input field for 'Organisation Identifier' which contains the text 'TheHiveProject'. Below the identifier field, there is a 'UUID' section with a text input field containing the UUID 'fd90c012-f47a-4c6b-81d6-5a5a8ed1ad40' and a 'Generate UUID' button. The form is styled with a light blue background and a dark blue header.

Posteriormente agregaremos un nuevo usuario asignando nombre, un logueo y le asignaremos el rol que tendrá.

Los roles disponibles son:


- **ReadOnly** → Solo permite la lectura de datos.
- **Analyze** → Permite la lectura de datos y el análisis de los mismos.
- **OrgAdmin** → Dentro de su organización, tiene todos los permisos.

En nuestro caso elegiremos **“OrgAdmin”**, ya que las operaciones que vamos a realizar requieren de todos los permisos.

The screenshot shows a web application interface for adding a user. At the top, there is a navigation bar with links: Home, Event Actions, Dashboard, Galaxies, Input Filters, Global Actions, Sync Actions, Administration, and Logs. On the left, a sidebar menu is visible with the following items: Add User (highlighted), List Users, Pending registrations, User settings, Set Setting, Contact Users, Add Organisation, List Organisations, Add Role, List Roles, and Server Settings &. The main content area is titled 'Admin Add User' and contains the following form fields:

- Email:** A text input field containing 'admin@thehive.local'.
- Set password:** A checkbox that is checked.
- Password:** A text input field with masked characters (dots).
- Confirm Password:** A text input field with masked characters (dots).
- Organisation:** A dropdown menu with 'TheHiveProject' selected.
- Role:** A dropdown menu with 'Org Admin' selected.
- NIDS SID:** A text input field containing '1'.

El siguiente paso será generar una “Auth Key” para el mismo, para ello, nos dirigimos a el siguiente apartado y generemos una nueva.

Auth keys 

« previous next »

+ Add authentication key

#

User


Auth Key

Expiration

Page 1 of 1, showing 0 records out of 0 total, starting on record 0, ending on 0


« previous next »

Se generará un formulario por si queremos añadir valores adicionales a la misma.

Add auth key 

Auth keys are used for API access. A user can have more than one authkey, so if you would like to use separate keys per tool that queries MISP, add additional keys. Use the comment field to make identifying your keys easier.

User

admin@thehive.local 

Comment

Allowed IPs

Expiration (keep empty for indefinite)

YYYY-MM-DD

☐ Read only (it will be not possible to do any change operation with this token)

Submit

Cancel

A continuación, se nos creará la misma, es importante copiarla ya que la usaremos posteriormente.

Auth key created



Please make sure that you note down the auth key below, this is the only time the auth key is shown in plain text, so make sure you save it. If you lose the key, simply remove the entry and generate a new one.

MISP will use the first and the last 4 characters for identification purposes.

ctUgujMEwaMau9159dmX7uHb98vwf8S1y7vc2kv9

I have noted down my key, take me back now

También podremos comprobar que el usuario ya tiene la clave asignada.

Authentication key Index

A list of API keys bound to a user.

« previous

next »

+ Add authentication key

| # | User | Auth Key |
|---|---------------------|---------------|
| 3 | admin@thehive.local | ctUg.....2kv9 |

Para integrar el servicio Misp con TheHive, nos dirigimos al apartado llamado con su propio nombre en el apartado de configuraciones del servidor en TheHive.

The screenshot shows the 'Gestión de plataforma' (Platform Management) section of TheHive. On the left is a sidebar with icons for navigation. The main area has a top navigation bar with tabs: 'Licencia', 'Estado', 'Marca', 'Corteza', 'MISP', and 'Aut'. The 'MISP' tab is selected and highlighted with a red underline. Below the tabs, there are two sections: 'General' and 'Servidores'. The 'General' section contains a field labeled '* Intervalo' with a value of '1' and a unit of 'hour'. The 'Servidores' section has a '+' icon to add new servers. Below this, there is a table with a header row labeled 'NOMBRE DEL SERVIDOR'.

| NOMBRE DEL SERVIDOR |
|---------------------|
|---------------------|

Aquí también podremos modificar diferentes valores generales de la conexión con Misp en caso de que sea necesario.

En nuestro caso, no será necesario, pero se puede incluir en alguna posible integración con Misp.

Una vez pulsado en el botón “+”, añadiremos el nombre que va a tener el servidor en la interfaz, la URL completa del servicio de Misp “(**Con la IP del contenedor**)” y la clave API que hemos copiado anteriormente.

Configurar el nuevo servidor

General

* Nombre del servidor

misp0

* URL del servidor

http://192.168.1.49/

* Clave API

.....

* Objetivo

Importar y exportar

Posteriormente, ejecutaremos una prueba de conexión con el otro contenedor Misp.

Lista de etiquetas permitidas

Lista de etiquetas prohibidas

Cancelar

Prueba de conexión al servidor

Agregar

La prueba debe devolver este resultado, de lo contrario será necesario verificar la configuración de la conexión o de los servicios.

success X

La configuración Misp ha sido probada con éxito

Una vez realizados estos pasos, ya tendremos integrados ambos servicios, lo podemos comprobar en el apartado de licencia y el símbolo del mensaje se debe poner en color verde.

Gestión de plataforma

Licencia

Estado

Marca

Corteza

MISP

Autenticación

SMTP

General

* Intervalo

1

hour

Servidores

NOMBRE DEL SERVIDOR

misp0

http://192.168.1.49/

5.2.15-3

Integración Splunk + (TheHive + Cortex)

Para continuar con la integración de las herramientas y construir un sistema robusto, vamos a entregar Splunk a TheHive y Cortex, para ello, en la interfaz de TheHive, pulsaremos en añadir una nueva organización.




El siguiente paso será asignar un nombre a la organización que estamos creando y establecer las reglas de compartición de tareas en **"autoshare"**.

The screenshot shows the 'Agregar una organización' (Add an organization) form. It has four main sections: 'Nombre' (Name) with the value 'SplunkConector', 'Descripción' (Description) with the value 'Conector con Splunk', 'Regla de compartición de tareas' (Task sharing rule) with the value 'autoShare', and 'Regla de compartición de observables' (Observable sharing rule) with the value 'autoShare'. Each section has a red underline under the input field.

Posteriormente podremos observar que la organización se ha creado correctamente.


| | | |
|--------------------------|--------|---|
| <input type="checkbox"/> | | NOMBRE ↑ |
| <input type="checkbox"/> | Activo | <div>A</div> admin Organizaciones vinculadas Ninguno |
| <input type="checkbox"/> | Activo | <div>S</div> SplunkConector Organizaciones vinculadas Ninguno |


El siguiente paso es crear un nuevo usuario para esta organización, para ello, pulsaremos en añadir un nuevo usuario.



Usuarios globales

→





<

Posteriormente agregaremos un nuevo usuario asignando nombre, un logueo y le asignaremos el rol que tendrá.

Los roles disponibles son:

- **ReadOnly** → Solo permite la lectura de datos.
- **Analyze** → Permite la lectura de datos y el análisis de los mismos.
- **OrgAdmin** → Dentro de su organización, tiene todos los permisos.

En nuestro caso elegiremos **“OrgAdmin”**, ya que las operaciones que vamos a realizar requieren de todos los permisos.

Añadir un usuario

Tipo

Normal

Los usuarios del servicio se utilizan esencialmente para los bots (autenticación de la clave API).

* Inicio de sesión

UsuarioSplunk@TheHive.com

* Nombre

UsuarioSplunk


Organizaciones

admin

☒ SplunkConector

org-admin

El siguiente paso será generar una **“Auth Key”** para el mismo, para ello, nos dirigimos a el siguiente apartado y generemos una nueva.



* Nombre

UsuarioSplunk

[Borrar](#)

Acceso

usuariosplunk@thehive.com

Email

Email

Tipo

Normal

Bloqueado

☐

MFA

No

Clave API

Crear

Revelar

Contraseña

A Continuación, se nos creará la misma, es importante copiarla ya que la usaremos posteriormente.

Clave API

MWEM7ab8JI0F7QQ/2CgunEV7ct4tMcDS


Renovar

Revelar

Splunk no cuenta con integración nativa con TheHive o Cortex, pero si que cuenta con una biblioteca de plugins que agregan al mismo múltiples funcionalidades, en esta misma, buscaremos el plugin de TheHive y Cortex y lo instalaremos.

Best Match Newest Popular

1 Apps

 **TheHive/Cortex** Open App

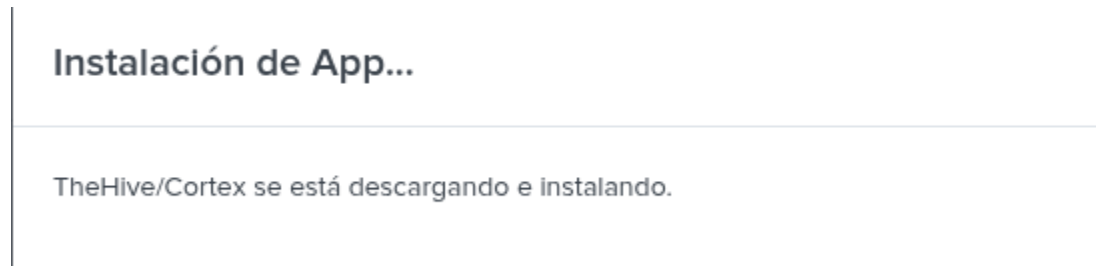
This TA allows to add interaction features between TheHive or Cortex (<https://www.strangebee.com/>) and Splunk. It allows to retrieve all kind of information from TheHive/Cortex and to perform actions on these instances using Splunk, from a search or from a predefined dashboard.

This TA is supporting only TheHive 5. For having an app supporting The... [More](#)

Category: [SIEM](#), [Ticketing](#) | Author: [Alexandre Demeyer](#) | Downloads: 2946 | Released: 24 days ago | Last Updated: 18 days ago |

[View on Splunkbase](#)

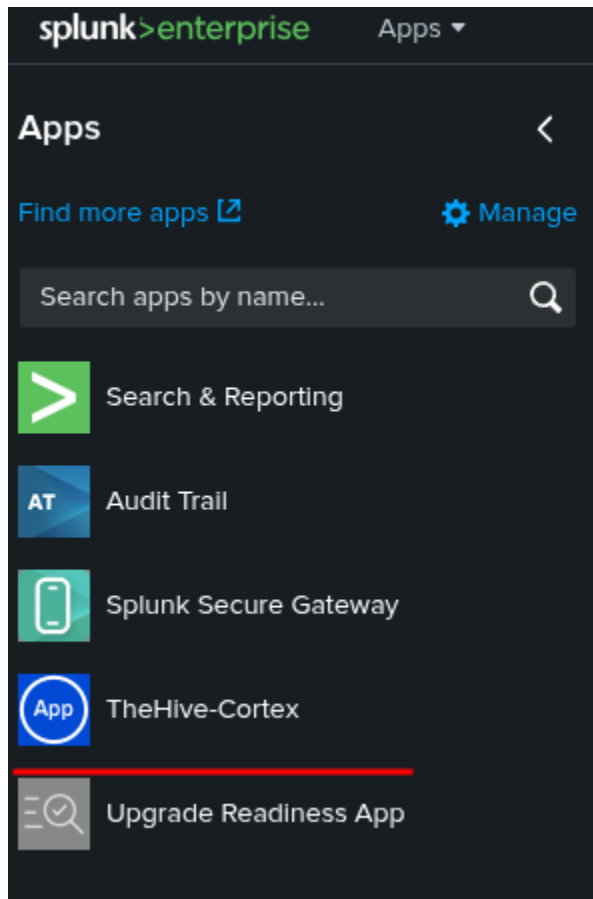
Esta es una muestra del panel que indica que el plugin se está descargando e instalando.



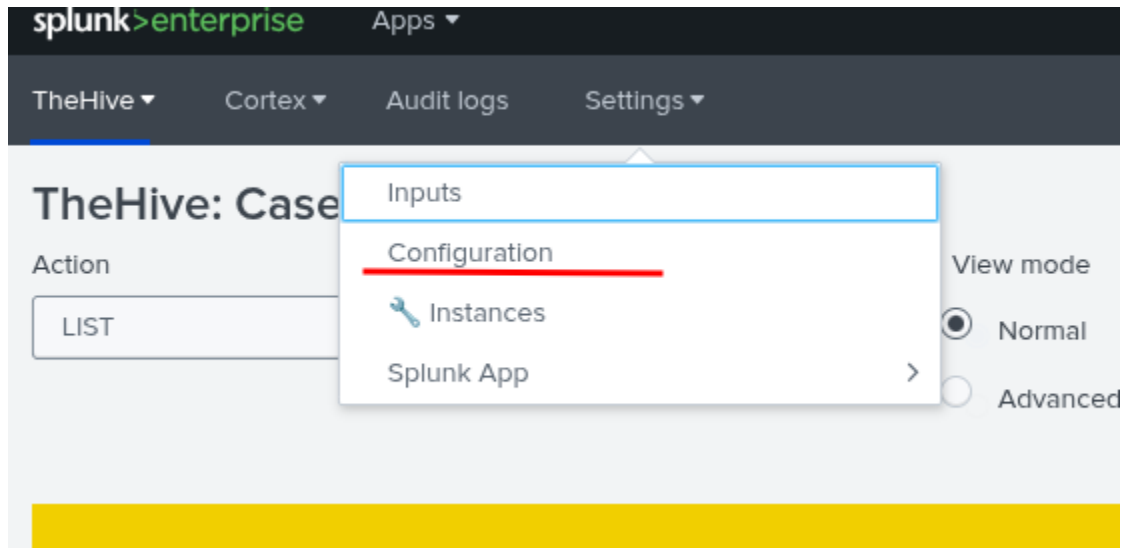
Una vez nos muestre el panel que indica que el plugin se ha descargado e instalado correctamente abriremos la aplicación desde la barra de aplicaciones de Splunk.



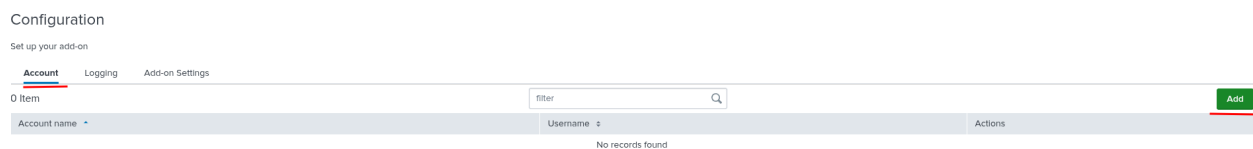
Para ello, nos dirigiremos al panel lateral izquierdo de la página principal de Splunk, donde encontraremos el icono junto con el nombre del plugin.



Una vez nos situemos dentro de la aplicación, deberemos dirigirnos al apartado de configuración general, donde accederemos a la propia configuración.



Posteriormente nos dirigiremos al apartado de cuentas, y pulsaremos en añadir una nueva cuenta.



Primero añadiremos la cuenta de TheHive, para ello tendremos que introducir el mismo usuario que hemos asignado a la cuenta que hemos creado previamente en TheHive y como contraseña configuraremos la clave API que hemos creado previamente a esta cuenta.

Add Account

| | |
|--------------|--|
| Account name | <input type="text" value="TheHiveInstance"/> |
| | Enter a unique name for this account. |
| Username | <input type="text" value="UsuarioSplunk"/> |
| | Enter the username for this account. |
| Password | <input type="password" value="....."/> |
| | Enter the password for this account. |

Posteriormente añadiremos la cuenta de Cortex, para ello tendremos que introducir el mismo usuario que hemos asignado a la cuenta que hemos creado previamente en Cortex y como contraseña configuraremos la clave API que hemos creado previamente a esta cuenta.

Add Account

| | |
|--------------|---|
| Account name | <input type="text" value="CortexInstance"/> |
| | Enter a unique name for this account. |
| Username | <input type="text" value="TheHive_Cortex_Connector"/> |
| | Enter the username for this account. |
| Password | <input type="password" value="....."/> |
| | Enter the password for this account. |

Una vez configuradas ambas, podremos verificar desde el propio panel de cuentas que se han creado correctamente.

Configuration

Set up your add-on

Account

Logging

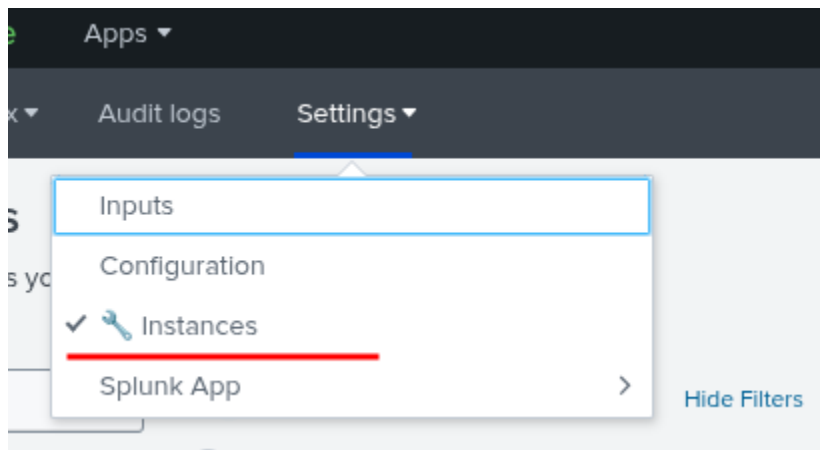
Add-on Settings

2 Items

filter

| Account name | Username |
|-----------------|--------------------------|
| CortexInstance | TheHive_Cortex_Connector |
| TheHiveInstance | UsuarioSplunk |

Ahora pasaremos a integrar los propios nodos dentro de Splunk, para ello, en el apartado de configuración general, seleccionaremos el apartado de instancias.



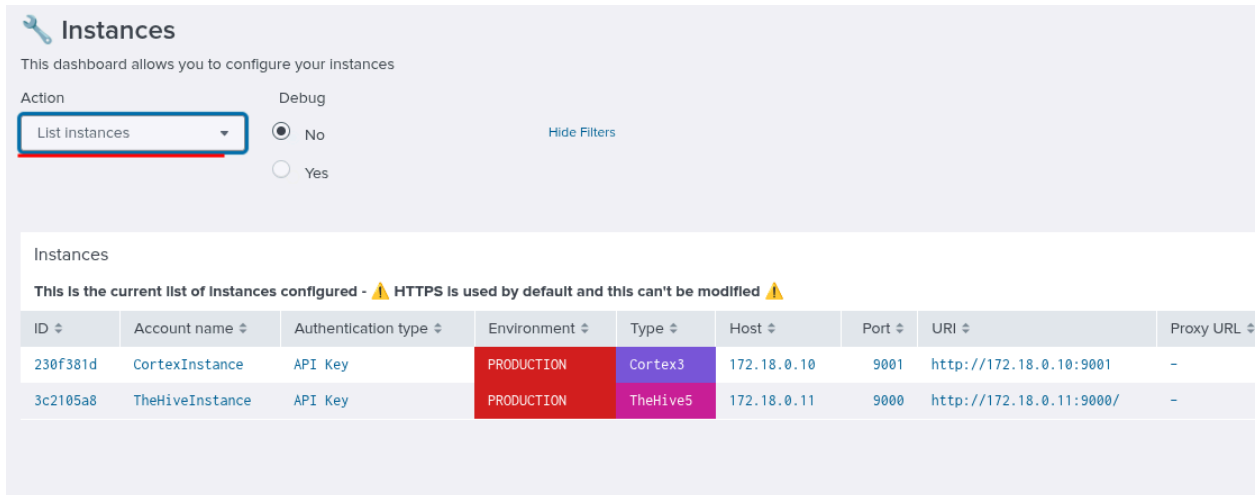
Para integrar Cortex en el apartado action seleccionamos “Add new instance”, seleccionamos como método de autenticación “**API Key**” de tipo “**Cortex3**” y añadiremos el nombre de la organización que hemos creado previamente, junto con IP, puerto y URL completo de acceso al contenedor Cortex.

The screenshot shows the 'Instances' configuration dashboard. The 'Action' dropdown is set to 'Add a new instance'. The 'Account name (Global accounts)' dropdown is set to 'CortexInstance'. The 'Environment' dropdown is set to 'PRODUCTION'. The 'Authentication type' dropdown is set to 'API Key'. The 'Type' dropdown is set to 'Cortex3'. The 'Organisation (Default:-)' text field contains 'TheHiveProject'. The 'Client Certificate (Default: -)' text field is empty. The 'Host' text field contains '172.18.0.10'. The 'Port' text field contains '9001'. The 'URI' text field contains 'http://172.18.0.10:9001'. The 'Comment' text field is empty. The 'Debug' radio buttons are set to 'No'.

Para integrar TheHive en el apartado action seleccionamos “Add new instance”, seleccionamos como método de autenticación “**API Key**” de tipo “**TheHive5**” y añadiremos el nombre de la organización que hemos creado previamente, junto con IP, puerto y URL completo de acceso al contenedor TheHive.

The screenshot shows the 'Instances' configuration dashboard. The 'Action' dropdown is set to 'Add a new instance'. The 'Account name (Global accounts)' dropdown is set to 'TheHiveInstance'. The 'Environment' dropdown is set to 'PRODUCTION'. The 'Authentication type' dropdown is set to 'API Key'. The 'Type' dropdown is set to 'TheHive5'. The 'Organisation (Default:-)' text field contains 'SplunkConnector'. The 'Client Certificate (Default: -)' text field is empty. The 'Host' text field contains '172.18.0.11'. The 'Port' text field contains '9000'. The 'URI' text field contains 'http://172.18.0.11:9000/'. The 'Comment' text field is empty. The 'Debug' radio buttons are set to 'No'.

Posteriormente ya se nos mostrarán ambas instancias conectadas.



Instances

This dashboard allows you to configure your Instances

Action: List Instances

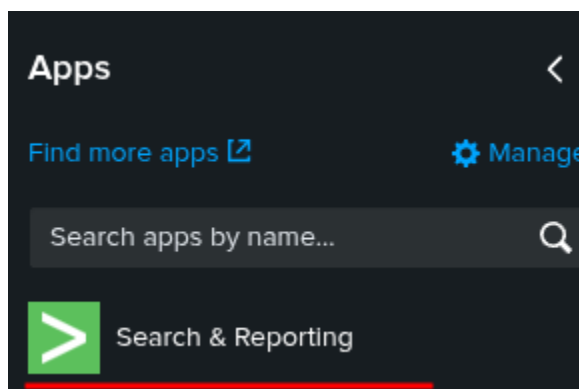
Debug: ☒ No ☐ Yes [Hide Filters](#)

Instances

This is the current list of Instances configured - ⚠️ HTTPS is used by default and this can't be modified ⚠️

| ID | Account name | Authentication type | Environment | Type | Host | Port | URI | Proxy URL |
|----------|-----------------|---------------------|-------------|----------|-------------|------|--------------------------|-----------|
| 230f381d | CortexInstance | API Key | PRODUCTION | Cortex3 | 172.18.0.10 | 9001 | http://172.18.0.10:9001 | - |
| 3c2105a8 | TheHiveInstance | API Key | PRODUCTION | TheHive5 | 172.18.0.11 | 9000 | http://172.18.0.11:9000/ | - |

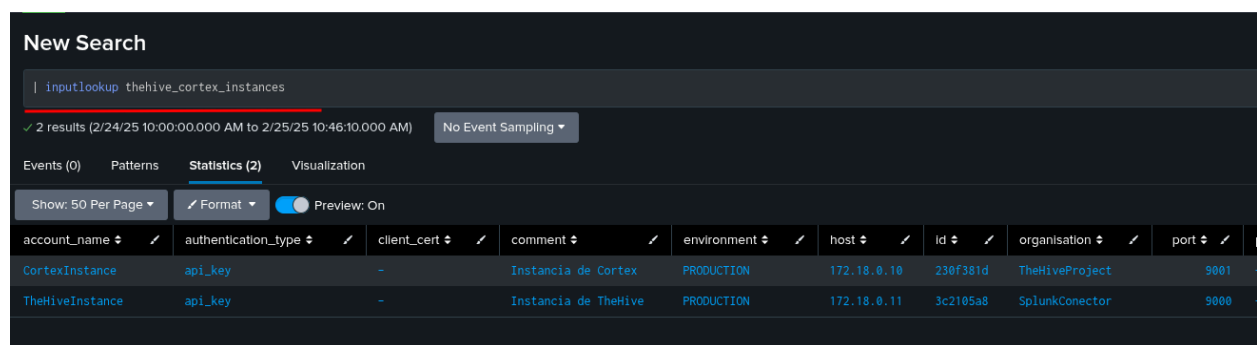
Para comprobar que la integración ha tenido éxito usaremos la APP integrada con Splunk “**Search & Reporting**”, la abriremos desde el panel de aplicaciones de Splunk.



Finalmente, dentro de esta aplicación, haremos la siguiente búsqueda.

"| inputlookup thehive_cortex_instances"

Esta devolverá todas las instancias conectadas a Splunk tanto de TheHive como de Cortex, si esta devuelve ambos nodos de forma correcta, la integración habrá sido exitosa.



The screenshot shows the Splunk search interface. At the top, the search bar contains the query `| inputlookup thehive_cortex_instances`. Below the search bar, it indicates '2 results' from '2/24/25 10:00:00.000 AM to 2/25/25 10:46:10.000 AM'. The 'Statistics (2)' tab is selected, showing a table with two rows of data. The table has columns for account_name, authentication_type, client_cert, comment, environment, host, id, organisation, and port. The first row is for 'CortexInstance' and the second for 'TheHiveInstance'.

| account_name | authentication_type | client_cert | comment | environment | host | id | organisation | port |
|-----------------|---------------------|-------------|----------------------|-------------|-------------|----------|----------------|------|
| CortexInstance | api_key | - | Instancia de Cortex | PRODUCTION | 172.18.0.10 | 230f381d | TheHiveProject | 9001 |
| TheHiveInstance | api_key | - | Instancia de TheHive | PRODUCTION | 172.18.0.11 | 3c2105a8 | SplunkConector | 9000 |

Integración Splunk + Misp

Para finalizar con la integración de las herramientas y construir un sistema robusto, vamos a entregar Splunk a Misp, logrando así un sistema completamente interconectado, descargamos el siguiente plugin en la tienda de Splunk.



MISP42

Install

With MISP42, connect your Splunk search head with your MISP instance(s). It is a versatile TA that acts as a wrapper of MISP API to either collect MISP information into Splunk (custom commands) or push information from Splunk to MISP (alert actions).

Category: [SIEM, Security, Fraud & Compliance](#) | Author: [Remi Seguy](#) | Downloads: 11365 | Released: 4 months ago |

Last Updated: 4 months ago | [View on Splunkbase](#)

Esta es una muestra del panel que indica que el plugin se está descargando e instalando.

Installing App...

MISP42 is being downloaded and installed.

Una vez nos muestre el panel que indica que el plugin se ha descargado e instalado correctamente deberemos reiniciar Splunk.

Restart Required

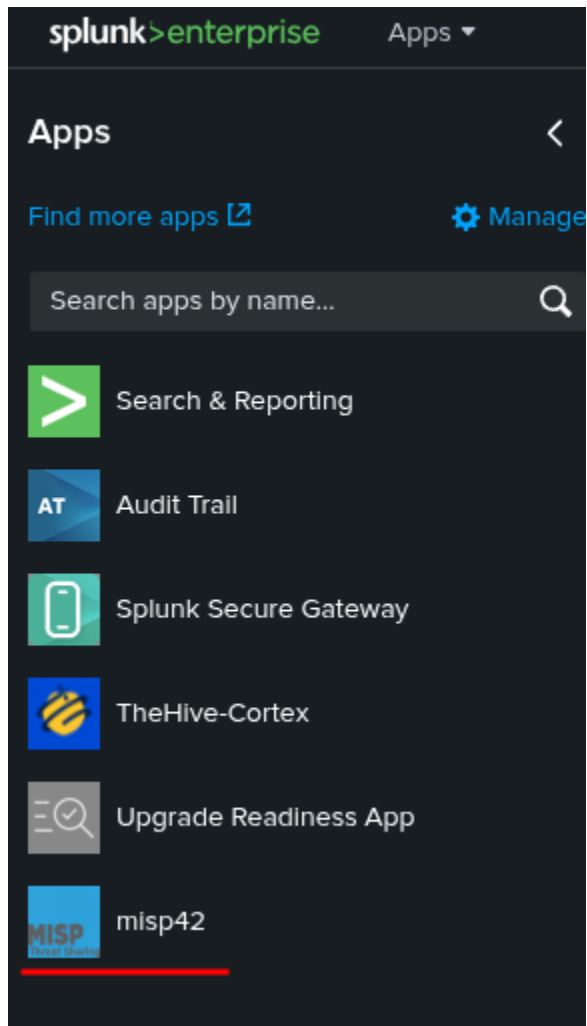


You must restart Splunk Enterprise to complete installation of MISP42.

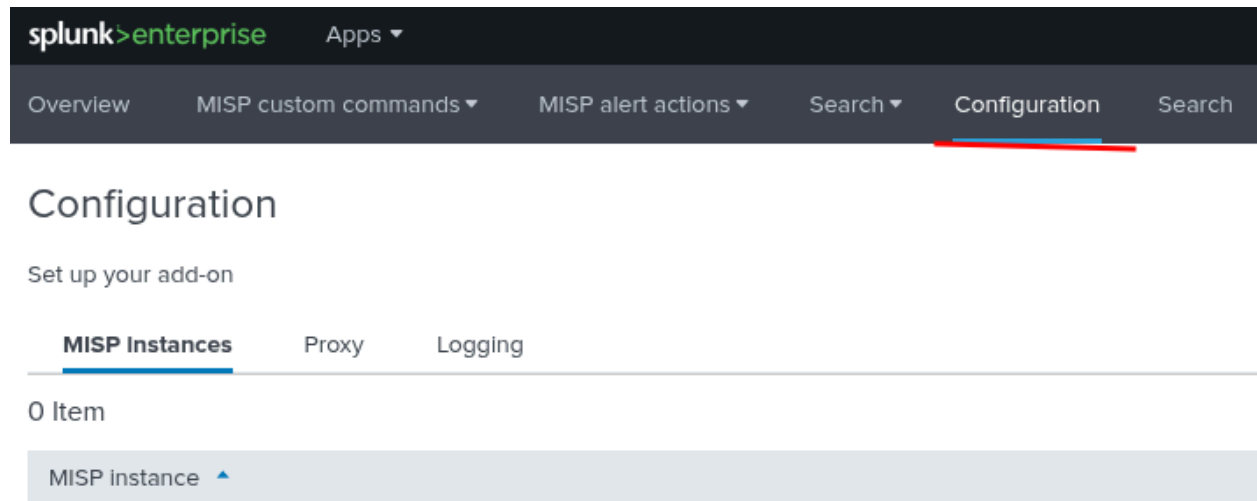
Restart Later

Restart Now

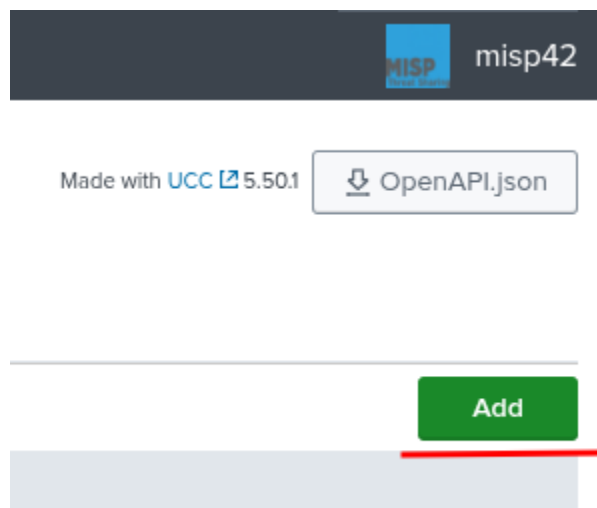
Posteriormente, nos dirigiremos al panel lateral izquierdo de la página principal de Splunk, donde encontraremos el icono junto con el nombre del plugin.



Una vez nos situemos dentro de la aplicación, deberemos dirigirnos al apartado de configuración general, donde accederemos a la propia configuración.



Aquí podremos observar que no tenemos ninguna instancia conectada, para conectar la misma, pulsaremos en el botón “Add”.



Ahora vamos a integrar Misp, para ello tendremos que introducir el nombre que va a tener la instancia en el panel, la URL de la instancia y como contraseña configuraremos la clave API que hemos creado previamente.

Add MISP instances

* MISP instance
Enter a unique name for this MISP instance.

* MISP url
provide base MISP URL starting with https://

* MISP API key
provide one authkey for the instance

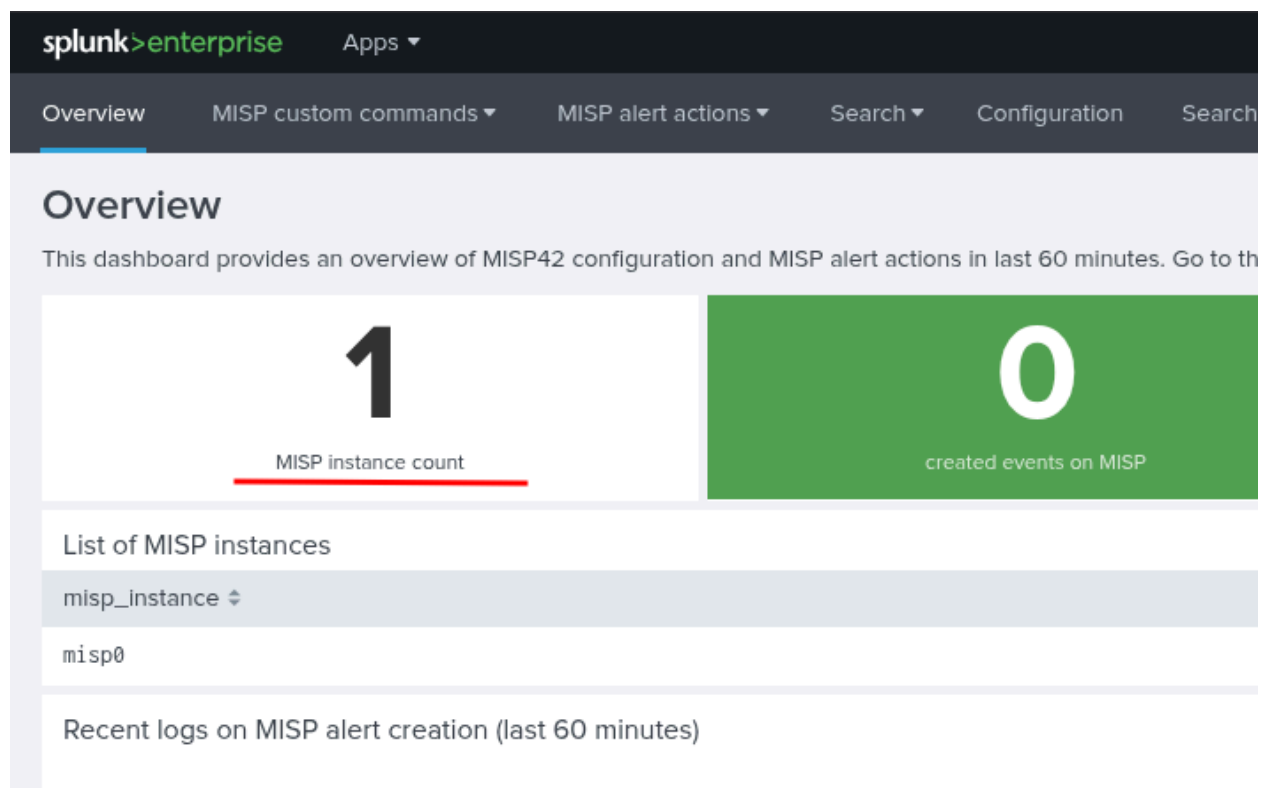
Posteriormente ya veremos el nodo agregado, solo queda comprobar que ha reconocido y ha establecido conexión con la instancia agregada.

Configuration

Set up your add-on

| MISP Instances | | Proxy | Logging |
|----------------|----------------------|--|---------|
| 1 Item | | <input type="text" value="Search..."/> | |
| MISP Instance | MISP url | | |
| misp0 | https://192.168.1.49 | | |

Para verificar lo anterior, debemos dirigirnos al dashboard principal de la aplicación, si vemos que el número de instancias ha aumentado a 1, significa que Splunk ha podido establecer conexión con la instancia de Misp, la integración ha tenido éxito.



Con esto hemos conseguido un sistema SOC totalmente integrado y capaz de colaborar entre sí, además, hemos conseguido la capacidad de gestionar todos los servicios de forma centralizada mediante Splunk.

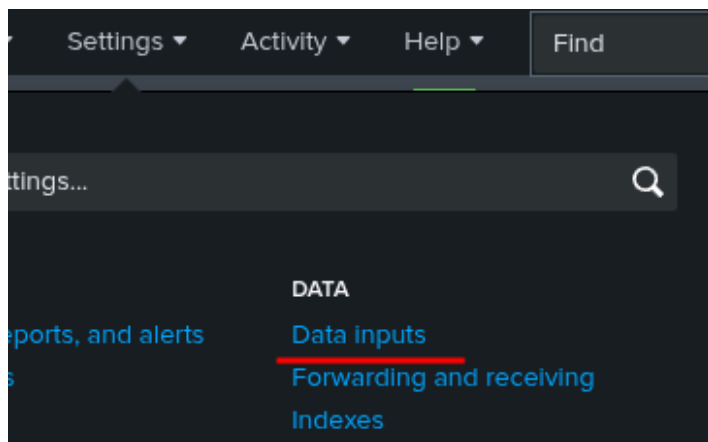
Fase 4: Prueba de integración en caso práctico

Creación de recolector HEC

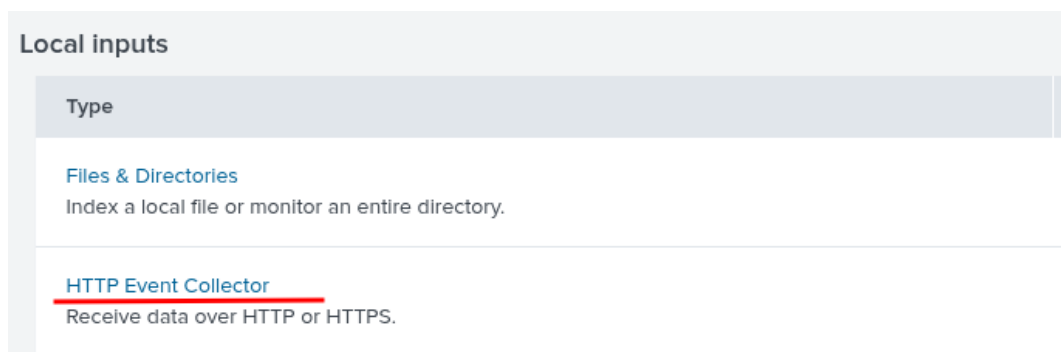
Un recolector HEC “(**HTTP Event Collector**)” en Splunk es una interfaz que permite recibir datos a través de solicitudes HTTP/HTTPS en formato JSON.

Es una forma eficiente de enviar eventos a Splunk desde aplicaciones, scripts o servicios sin necesidad de usar forwarders tradicionales.

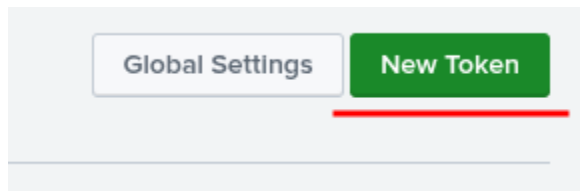
En nuestro caso, lo utilizaremos para recibir datos de las aplicaciones que configuraremos posteriormente, el primer paso para configurarlo es dirigirse a “**Setting > Data inputs**”.



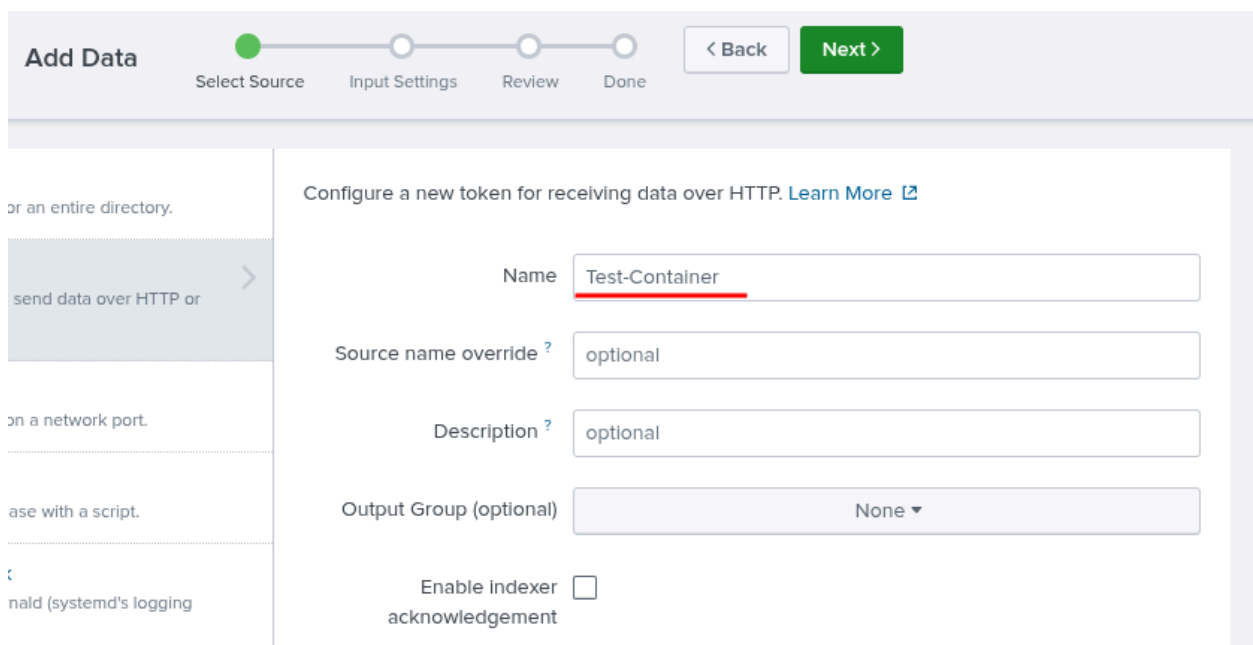
Posteriormente lo seleccionaremos entre la lista de “**Data inputs**” disponibles.



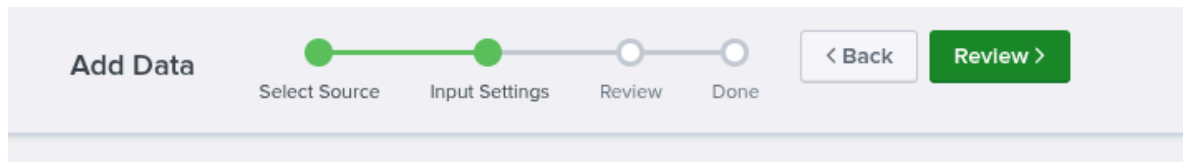
Posteriormente pulsaremos en el apartado **“New Token”** en la parte superior derecha de la pantalla.



Seguidamente empezaremos con la creación del propio HEC, el primer paso será añadir un nombre personalizado al mismo.

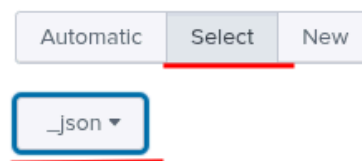
A screenshot of the 'Add Data' wizard in a web application. The wizard has four steps: 'Select Source', 'Input Settings', 'Review', and 'Done'. The 'Input Settings' step is currently active, indicated by a green dot. The main content area shows the configuration for a new token for receiving data over HTTP. It includes a 'Name' field with the value 'Test-Container', a 'Source name override' field with the value 'optional', a 'Description' field with the value 'optional', and an 'Output Group (optional)' dropdown menu set to 'None'. There is also a checkbox for 'Enable indexer acknowledgement' which is currently unchecked. A 'Learn More' link is visible next to the title 'Configure a new token for receiving data over HTTP'. On the left side, there is a sidebar with a list of data sources, including 'send data over HTTP or', 'on a network port.', 'ase with a script.', and 'nald (systemd's logging'.

El siguiente paso será elegir el tipo de datos que admitirá, para que la configuración sea correcta elegiremos tipo “_json”.



input parameters for this data input as follows:

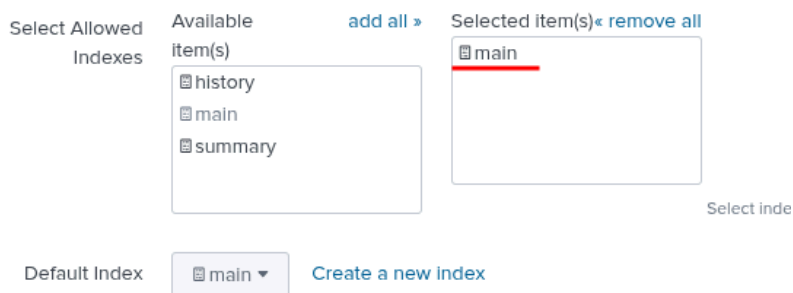
of the default fields that the Splunk coming data. It tells the Splunk platform got, so that the Splunk platform can tly during indexing. And it's a way to that you can search it easily.



También será necesario asignar un índice para filtrar los datos posteriormente.

Index

The Splunk platform stores incoming data as events in the selected index. Consider using a "sandbox" index as a destination if you have problems determining a source type for your data. A sandbox index lets you troubleshoot your configuration without impacting production indexes. You can always change this setting later. [Learn More](#)



Finalmente se nos mostrará un resumen de la configuración que hemos realizado en el HEC.

The screenshot shows the 'Add Data' wizard in the Review stage. At the top, a progress bar indicates the steps: Select Source, Input Settings, Review (current), and Done. The main content area is titled 'Review' and lists the following configuration details:

- Input Type Token
- Name Test-Container
- Source name override N/A
- Description N/A
- Enable indexer acknowledg No
- Output Group N/A
- Allowed indexes

main
- Default index main
- Source Type _json
- App Context search

Posteriormente se nos mostrará el token asociado al HEC, es importante copiarlo ya que será utilizado posteriormente.

The screenshot shows the 'Add Data' wizard in the 'Done' stage. The progress bar at the top shows all steps completed, with 'Done' marked with a checkmark. Navigation buttons '< Back' and 'Next >' are visible. A green checkmark icon is followed by the text 'Token has been created successfully.' and a link to 'Configure your inputs by going to Settings > Data Inputs'. Below this, the 'Token Value' is displayed as '94afd92b-2eff-4c59-97bb-ab310ca9' in a text box. At the bottom, there are five action buttons with descriptions:

- Start Searching**: Search your data now or see [examples and tutorials](#).
- Extract Fields**: Create search-time field extractions. [Learn more about fields](#).
- Add More Data**: Add more data inputs now or see [examples and tutorials](#).
- Download Apps**: Apps help you do more with your data. [Learn more](#).
- Build Dashboards**: Visualize your searches. [Learn more](#).

Despliegue de entorno de pruebas

Una vez tengamos el recolector HEC configurado, el siguiente paso será crear un contenedor que envíe información al mismo, en nuestro caso utilizaremos un contenedor de prueba que envíe eventos al HEC.

Para, implementaremos la siguiente configuración en un archivo **“.yml”** diferente al del proyecto principal.

```
! test-container.yml
1  services:
2    sender:
3      build: .
4      container_name: splunk_event_sender
5      restart: always
6      networks:
7        - SOC_NET
8
9  networks:
10   SOC_NET:
11     driver: bridge
12
```

Este requerirá de un archivo **“Dockerfile”** el cual construirá la imagen del sistema principal.

```
Dockerfile
1  FROM debian:latest
2
3  # Instalar Curl
4  RUN apt-get update && apt-get install -y curl
5
6  # Copiar el script al contenedor
7  COPY enviar_eventos.sh /usr/local/bin/enviar_eventos.sh
8
9  # Dar permisos de ejecución
10 RUN chmod +x /usr/local/bin/enviar_eventos.sh
11
12 # Ejecutar el script al iniciar el contenedor
13 CMD ["/usr/local/bin/enviar_eventos.sh"]
14
```

Para enviar información al recolector HEC, para ello desarrollaremos un script en el lenguaje “Bash” el cual enviará información a modo de prueba, para que este funcione es importante cambiar la variable “TOKEN” por el token generado anteriormente.

```
$ enviar_eventos.sh
1  #!/bin/bash
2
3  SPLUNK_URL="http://splunk:8088/services/collector"
4  TOKEN="94afd92b-2eff-4c59-97bb-ab310ca977b3" # Cambia este token por tu token de HEC
5
6  while true; do
7      echo "Enviando evento a Splunk..."
8      curl -k "$SPLUNK_URL" \
9          -H "Authorization: Splunk $TOKEN" \
10         -H "Content-Type: application/json" \
11         -d '{"event": "Evento desde contenedor", "sourcetype": "manual", "index": "main"}'
12
13     sleep 60
14 done
15
```

Tras levantar este nuevo contenedor, con la aplicación “Search & Reports” ejecutaremos la siguiente búsqueda para comprobar que Splunk está recibiendo la información.

New Search

index=* | sort -_time | head 20

✓ 4 events (2/25/25 9:00:00.000 AM to 2/26/25 9:19:18.000 AM) No Event Sampling ▼

Events (4) Patterns Statistics Visualization

✓ Timeline format ▼ — Zoom Out + Zoom to Selection × Deselect

Format Show: 20 Per Page View: List

| | i | Time | Event |
|--|---|---------------------------|--|
| | > | 2/26/25 9:08:42.000 AM | Evento desde contenedor host = splunk:8088 source = http:prueba python sourcetype = manual |
| | > | 2/26/25 9:07:42.000 AM | Evento desde contenedor host = splunk:8088 source = http:prueba python sourcetype = manual |
| | > | 2/26/25 8:49:47.000 AM | Este es un evento de prueba host = localhost:8088 source = http:prueba python sourcetype = manual |
| | > | 2/26/25 8:46:03.000 AM | Este es un evento de prueba host = localhost:8088 source = http:prueba python sourcetype = manual |

SELECTED FIELDS
a host 2
a source 1
a sourcetype 1

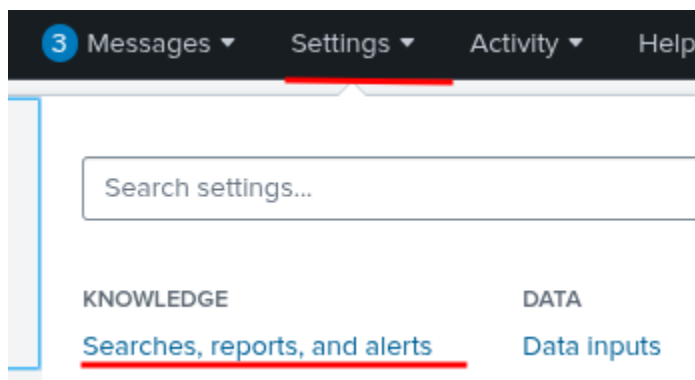
INTERESTING FIELDS
a index 2
linecount 1
a punct 2
a splunk_server 1

+ Extract New Fields

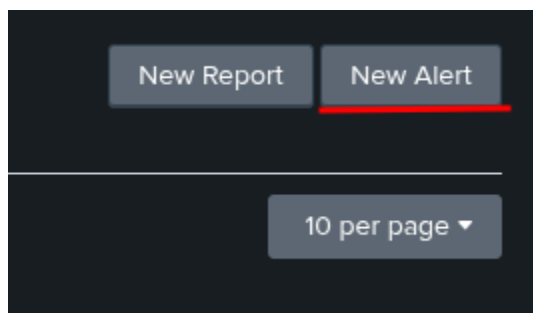
Creación de sistema de respuesta automático

Una vez que tengamos la infraestructura previa desplegada correctamente, pasaremos a crear respuestas automáticas para que cuando Splunk detecte que se reciba información a través del HEC realice acciones de forma automática.

El primer paso para configurarlo es dirigirse a **“Setting > Data inputs”**.



Posteriormente, pulsaremos en el botón **“New Alert”** en la parte superior derecha de la pantalla.



El siguiente paso consistirá en configurar los ajustes generales de esta nueva alarma, lo esencial es asignarle un nombre y una descripción a la misma.

Posteriormente, introduciremos la búsqueda que anteriormente usamos para comprobar si Splunk había datos, para que la alarma se active cuando se reciba algún dato nuevo.

Posteriormente seleccionaremos la App **“Search & Reports”** integrada en Splunk como App que se utilizará para realizar la búsqueda en los datos.

El siguiente paso es seleccionar los permisos que tendrá la alerta, en nuestro caso será **“Shared in App”**, para que la alerta tenga los permisos necesarios para comunicarse con otras apps o Plugins de Splunk.

Finalmente seleccionaremos cada cuanto tiempo se realizará la búsqueda, en nuestro caso será en **“Tiempo real”**, por lo que la observabilidad será inmediata. También seleccionaremos un tiempo de expiración de la alerta en caso de que sea necesario.

Create Alert [X]

Settings

Title: Test_Container_Actions

Description: Acciones que se van a desarrollar en base a los inputs del test_container.

Search: `index=* | sort -_time | head 20`

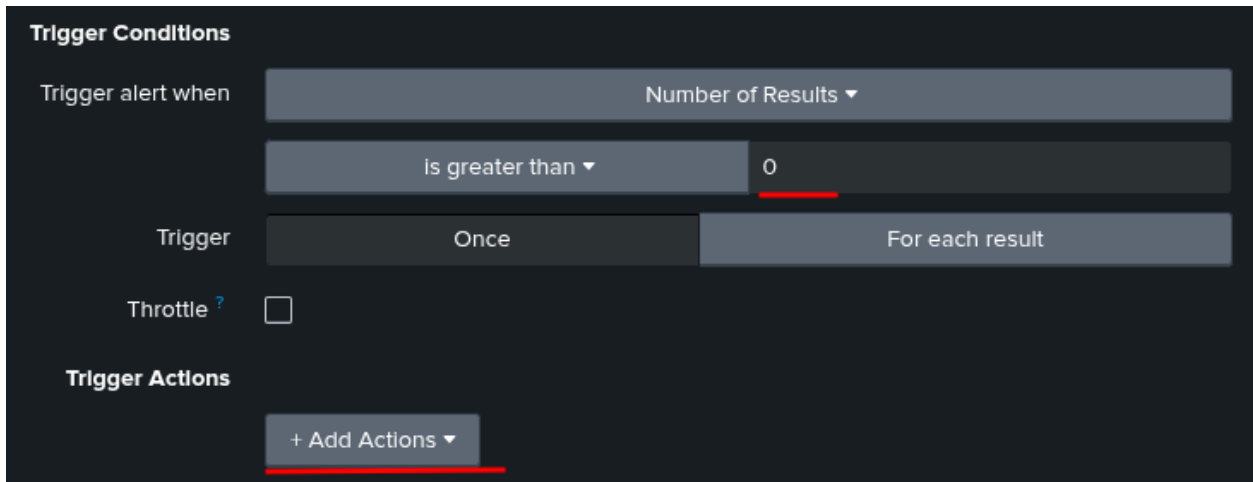
App: Search & Reporting (search) ▼

Permissions: Private Shared in App

Alert type: Scheduled Real-time

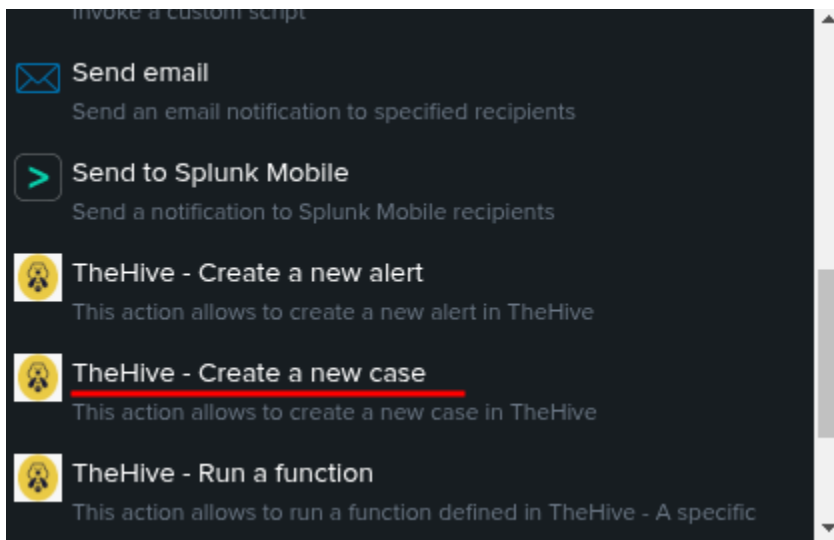
Expires: 500 day(s) ▼

Finalmente, configuraremos que el tigger de la alerta se active cada vez que haya un resultado nuevo en la búsqueda, ahora queda configurar las acciones que este hará para ello, pulsaremos en **“Add actions”**.



The screenshot shows the 'Trigger Conditions' configuration panel. It includes a dropdown for 'Trigger alert when' set to 'Number of Results', a comparison operator 'Is greater than' with a value of '0', and a 'Trigger' section with 'Once' and 'For each result' options. A 'Throttle' checkbox is present and unchecked. Below these is the 'Trigger Actions' section with a '+ Add Actions' button.


Primero configuraremos las acciones que hará en TheHive, para ello, seleccionamos **“TheHive - Create new case”**.



The screenshot shows a dropdown menu for selecting actions. The options are: 'Send email', 'Send to Splunk Mobile', 'TheHive - Create a new alert', 'TheHive - Create a new case' (which is highlighted with a red underline), and 'TheHive - Run a function'.

Posteriormente seleccionaremos la instancia en la que queremos que se ejecuten las acciones, además de asignar un **“ID”** único a los casos que son generados por Splunk, además de configurar un título y descripción de la alerta creada.

When triggered

 TheHive - Create a new case

Instance

SplunkConector

X

Indicate which instance to use (Set the 'id' provided under 'Instances'). You can use '<default>' to set automatically the ID to the default set parameter in the configuration page.

Case mode

ES notable

Alert action

Unique ID field

1

A field name that contains a unique identifier specific to the source event

Case Template

Prueba

The case template to use for imported alerts

Source

splunk

The alert source. Defaults 'splunk'

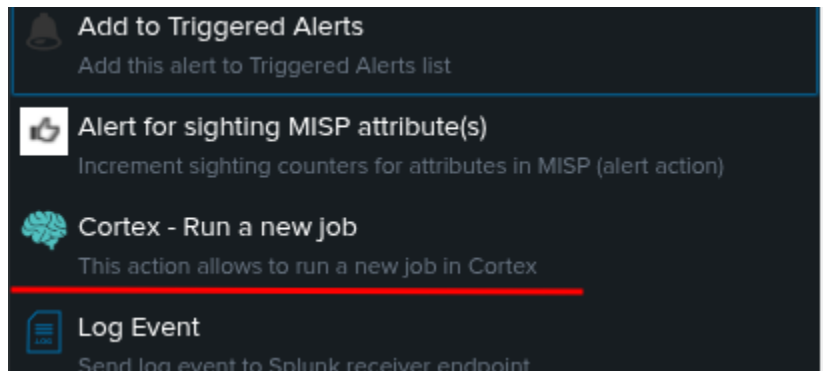
Timestamp field

A field name that contains a valid time-stamp. if not provided, default to now()

Title

Esto es una prueba de que funcioi

Proseguiremos por configurar las acciones que hará en Cortex, para ello, seleccionamos **“Cortex - Run a New Job”**.



Aquí indicaremos el ID de instancia que se nos ha generado en la integración de Cortex con Splunk, además de seleccionar el analizador que utilizaremos sobre los datos **“(En esto entraremos en más profundidad posteriormente)”**.

When triggered Remove

Cortex - Run a new job

Cortex Instance ID * 230f381d
Indicate which instance to use (Set the "id" provided under "Instances")

Data field name * URL
Name of the field to use as "data"

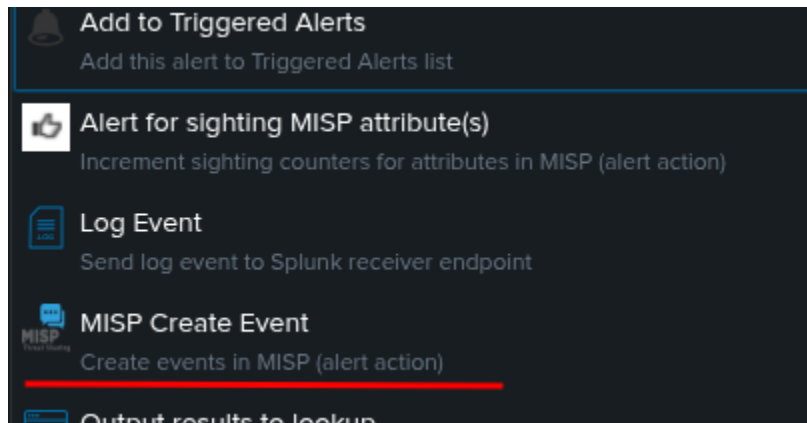
Datatype field name * URL
Name of the field to use as "datatype"

Analyzers * Urlscan.io
Indicate which analyzers to use by specifying the name of all analyzers separated by ","

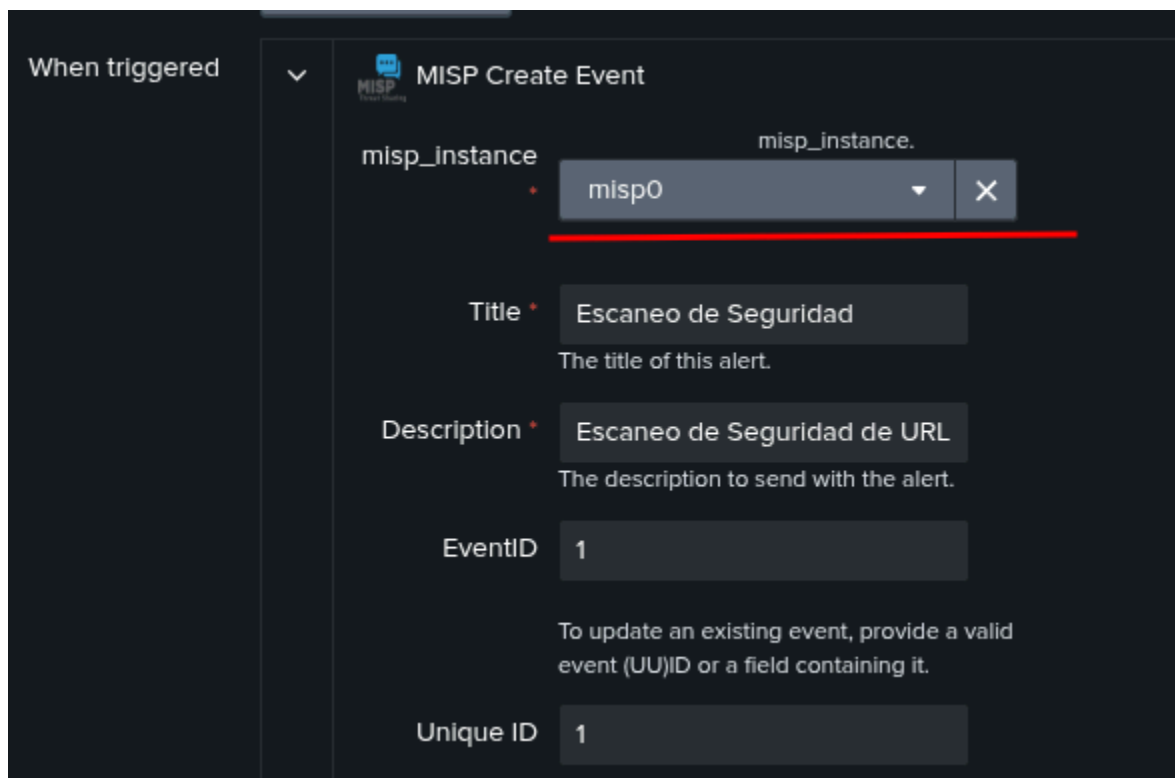
TLP: * AMBER
Select the TLP level of the created job. Default is TLP:AMBER

PAP: * AMBER
Permissible Action Protocol. Default to PAP:AMBER

Proseguiremos por configurar las acciones que hará en Cortex, para ello, seleccionamos **“Misp Create Event”**.



Seleccionamos la instancia de Misp que hemos integrado anteriormente con Splunk, y añadimos el título y descripción que tendrá el evento que se cree además de un **“ID”** único.



When triggered

▼ MISP Create Event

misp_instance misp_instance.
misp0

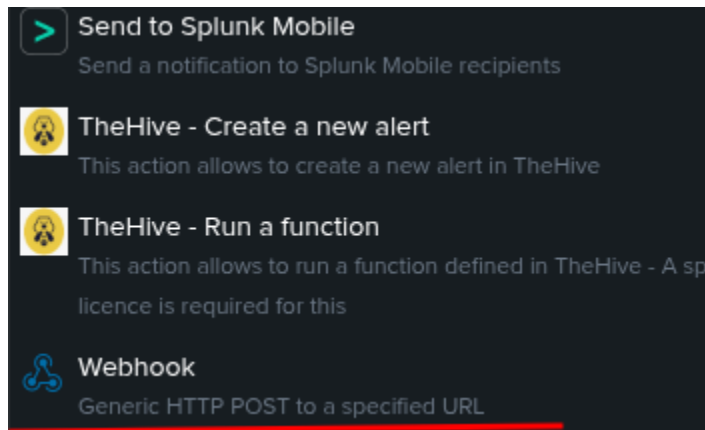
Title * Escaneo de Seguridad
The title of this alert.

Description * Escaneo de Seguridad de URL
The description to send with the alert.

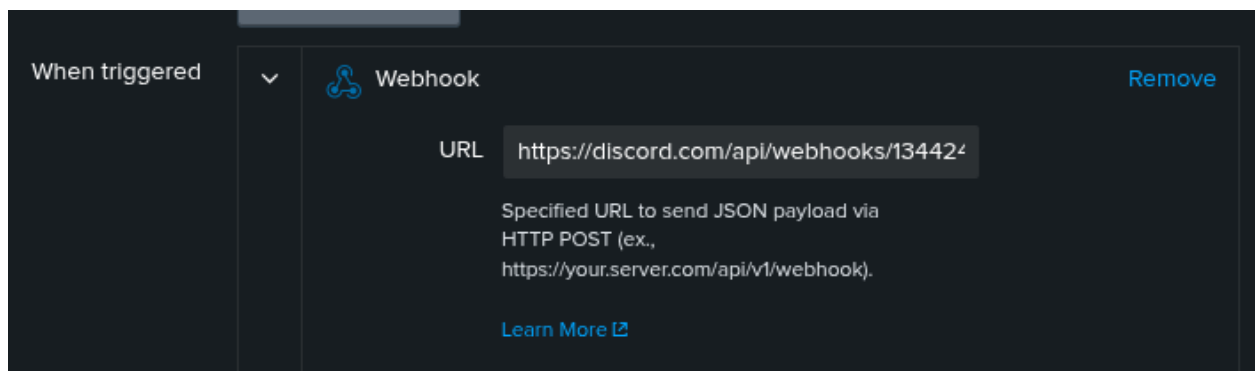
EventID 1
To update an existing event, provide a valid event (UU)ID or a field containing it.

Unique ID 1

Finalmente, haremos que cuando se dispare la alerta se envíe un mensaje automático a Discord para avisarnos al instante de que ha pasado algo. para ello, seleccionamos **“WebHook”**.

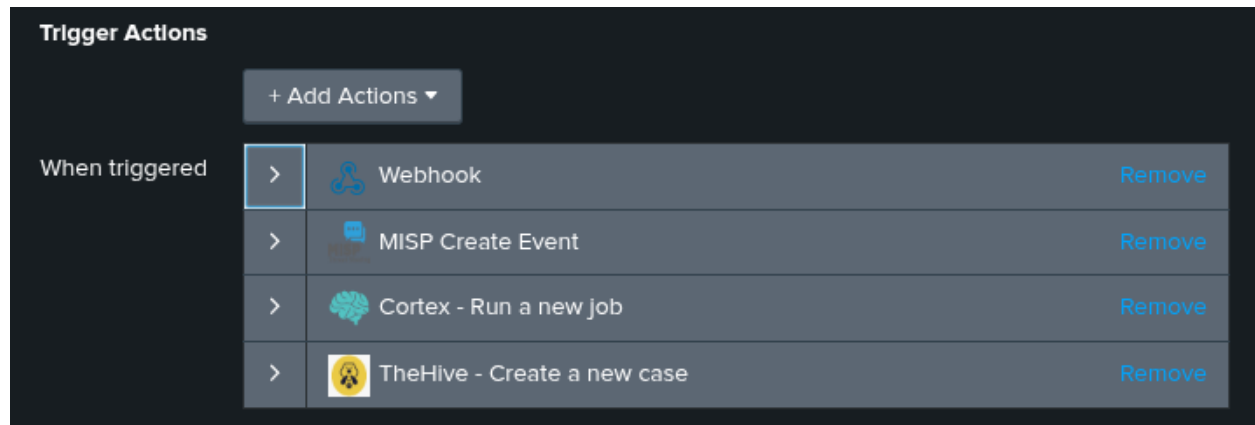


El siguiente paso será pegar el URL del WebHook que hayamos creado en Discord, si en tu caso no lo tienes creado, posteriormente se indicará como hacerlo.



Si en tu caso no lo tienes creado de forma previa, posteriormente se indicará como hacerlo.

Finalmente, estas serán todas las acciones que se realizarán cada vez que Splunk reciba información a través de el recolector HEC configurado previamente.



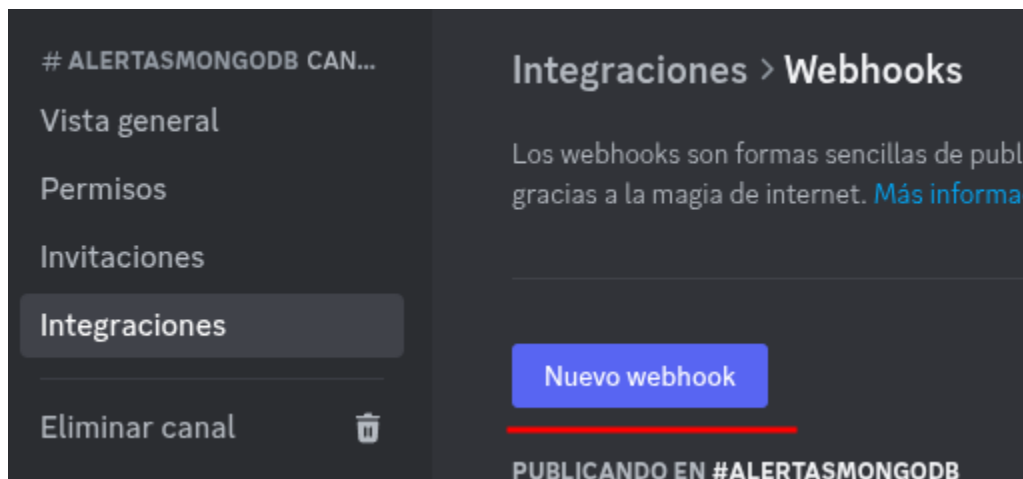
Con estas configuraciones conseguiremos que cada vez que Splunk reciba un dato, cree un caso en TheHive, donde los integrantes de la organización pueden ver que está pasando, este mismo a su vez creará un **"Job"** en Cortex **"(posteriormente configuraremos el mismo)"** y Misp comparara el supuesto ataque con una base de datos de gran tamaño para determinar si es realmente un ataque **"(posteriormente configuraremos esta característica)"**.

A su vez, Splunk enviará un mensaje a Discord para que nosotros mismos estemos al tanto de la situación.

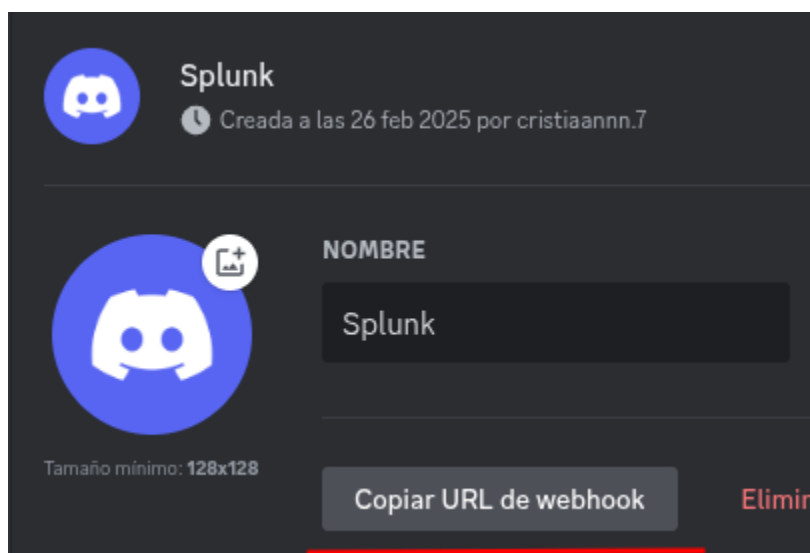
Este es tan solo un ejemplo de configuración para comprobar que la integración ha tenido éxito, pero se pueden analizar datos de todo tipo y reaccionar a ellos en tiempo real, Splunk es capaz de recibir datos y métricas de múltiples tipos, y con las configuraciones apropiadas, este sistema es capaz de analizar y reaccionar a cualquier información que reciba.

Configuraciones específicas

Para que este sistema funcione, debemos configurar algunos aspectos específicos y que pueden variar según el sistema que se quiera analizar, para empezar crearemos el WebHook necesario para recibir las comunicaciones de Splunk, para ello, nos dirigimos a **"Integraciones > Nuevo WebHook"** en las configuraciones de un canal propio de discord.



Posteriormente, copiaremos el URL del WebHook y lo pegaremos en la configuración anteriormente mostrada en Splunk.



Ahora configuraremos un analizador en Cortex que analiza la información enviada por el contendor de testeo, para ello, necesitaremos de los servicios de “**UrlScan.io**”, una página encargada de analizar URL y determinar si son peligrosos, nos tendremos que registrar en la misma.

Información del usuario

Nombre (Requerido)

Apellido (Requerido)

Dirección de correo electrónico (Requerido)

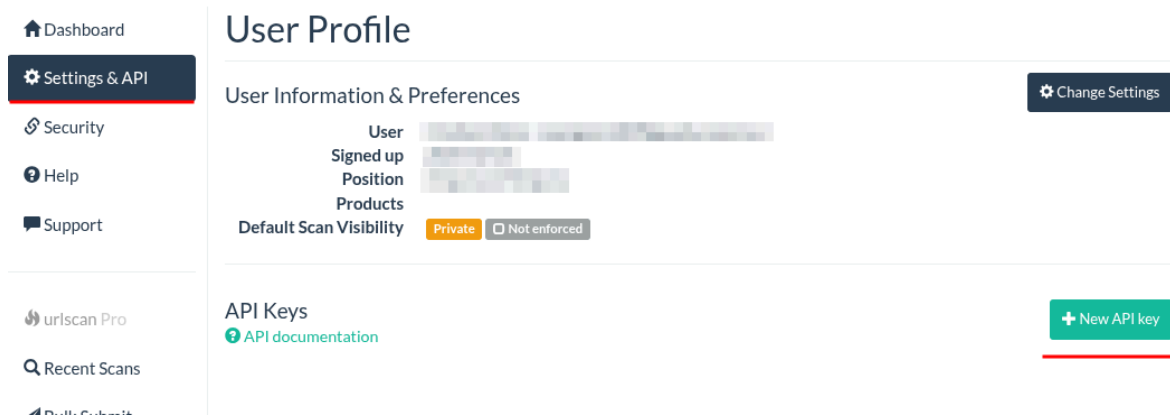
Atención: Debe usar una dirección de correo electrónico válida para recibir el correo de activación. Por favor **no usar** una dirección outlook.com o hotmail.com.

Contraseña (Requerido)

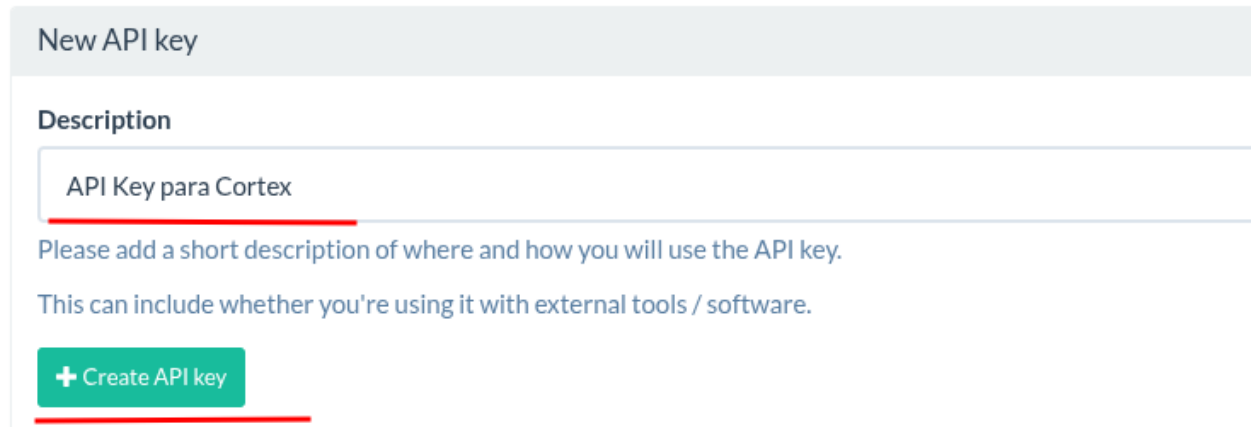
Atención: La contraseña debe tener al menos 8 caracteres, contener un dígito, un carácter en mayúscula, un carácter en minúscula y un carácter especial.

Para ello, introduciremos nuestras credenciales en el apartado de registro de la propia página, tales como nombre, apellido, dirección de correo electrónico y una contraseña segura.

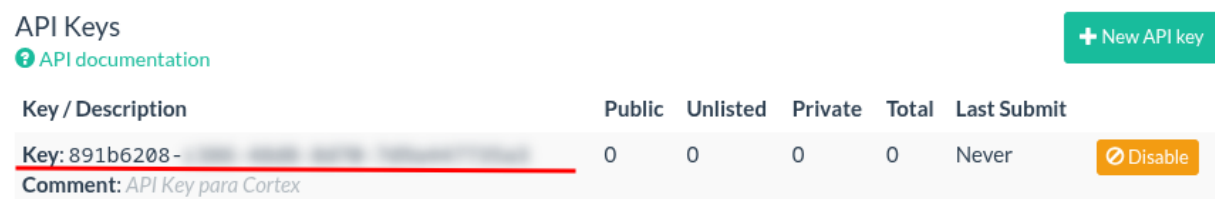
Posteriormente, en el apartado de configuración de perfil, crearemos una nueva **"API key"**, para ello, pulsaremos en **"New API Key"**.



El siguiente paso es añadir una descripción de la misma y crearla.



Finalmente se nos revelará la misma, debemos copiarla para su posterior uso.



| Key / Description | Public | Unlisted | Private | Total | Last Submit | |
|--|--------|----------|---------|-------|-------------|---------|
| Key: 891b6208- Comment: API Key para Cortex | 0 | 0 | 0 | 0 | Never | Disable |

A Continuación, pasaremos a configurar el analizador en Cortex. Un analizador es un módulo que permite ejecutar tareas de análisis sobre datos de seguridad, como IPs, dominios, hashes, URLs, archivos, entre otros. Estos analizadores pueden utilizar servicios externos o herramientas internas para enriquecer información de amenazas.

En nuestro caso utilizaremos los servicios de “**UrlScan.io**”, como hemos indicado anteriormente.

Organization: TheHiveProject

Users

Analyzers Config

Analyzers

Responders Config

Responders

Available analyzer configurations (115)

Q

URLScan

| Options | Configuration |
|---------|--|
| 1 | Urlscan.io |
| Option | <div>✖ key: API key for Urlscan.io</div> |

Para ello, editaremos el analizador y pegaremos el “**API Key**” generado anteriormente.

Edit configuration: Urlscan.io

key

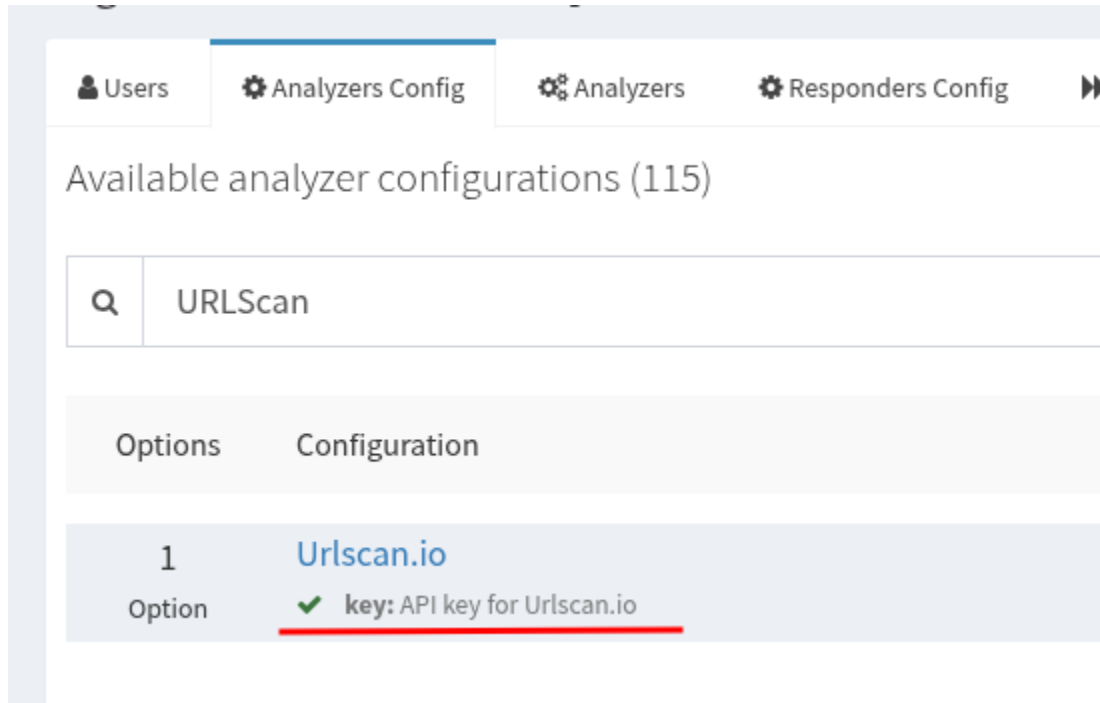
891b6208-c386-48d8-8d70-7d9a447735a3

API key for Urlscan.io

Cancel

Save

Finalmente, podremos comprobar que la conexión con la página es correcta cuando el tick del analizador se ponga en color verde.



The screenshot shows a web application interface with a top navigation bar containing four tabs: 'Users', 'Analyzers Config' (selected), 'Analyzers', and 'Responders Config'. Below the tabs, the text 'Available analyzer configurations (115)' is displayed. A search bar with a magnifying glass icon contains the text 'URLScan'. Below the search bar, there is a table with two columns: 'Options' and 'Configuration'. The table has one row with the following data:

| Options | Configuration |
|-------------|---|
| 1 Option | Urlscan.io ✓ key: API key for Urlscan.io |

Posteriormente pasaremos a configurar un responder para el analizador creado previamente, en Cortex, un responder es un módulo que permite tomar acciones sobre los datos analizados, los responders permiten ejecutar acciones automáticas para contener amenazas, mitigar riesgos o notificar equipos de seguridad.

En nuestro caso, utilizaremos **“UmbrellaBlacklister”**, el cual pondrá en una lista negra a la URL que se considere como peligrosa por el analizador anterior.

👤 Users ⚙️ Analyzers Config ⚙️ Analyzers ⚙️ Responders Config ▶▶ Responders

Available responder configurations (40)

Q

UmbrellaBlacklister

| Options | Configuration |
|---------|---|
| 1 | UmbrellaBlacklister |
| Option | <div><div>✖</div><div>integration_url: Custom integration url</div></div> |

Para ello, agregaremos la URL del propio contenedor que va a enviar los datos a Splunk, esto solo lo haremos a modo de prueba, en entornos de producción esta no es la configuración ideal.

Edit configuration: UmbrellaBlacklister

integration_url

http://172.18.0.3

Custom integration url

Cancel

Save

Finalmente, podremos comprobar que la conexión con el otro contenedor es correcta cuando el tick del analizador se ponga en color verde.

Q

UmbrellaBlacklister

Options

Configuration

1

UmbrellaBlacklister

Option

✓ integration_url: Custom integration url

Ahora pasaremos a habilitar el propio responder, para ello, lo buscaremos en el apartado de “Responders” el pulsaremos en el botón “Enable”.

Users

Analyzers Config

Analyzers

Responders Config

Responders

Available responders (117)

Q

Umbrella

Responders

Umbrella_Blacklister_1_1

Version: 1.1 Author: Kyle Parrish License: AGPL-V3 Type: Docker

Add domain to Umbrella blacklist via Enforcement API.

También podremos configurar algunos parámetros opcionales, en nuestro caso no será necesario.

Enable responder Umbrella_Blacklister_1_1

Base details

Name Umbrella_Blacklister_1_1

Configuration

[Apply defaults](#)

integration_url *
Custom integration url

Options

[Apply defaults](#)

Enable TLP check ☒ True ☐ False **Max TLP**

Enable PAP check ☒ True ☐ False **Max PAP**

HTTP Proxy

HTTPS Proxy

CA Certs

Job timeout

Rate Limiting

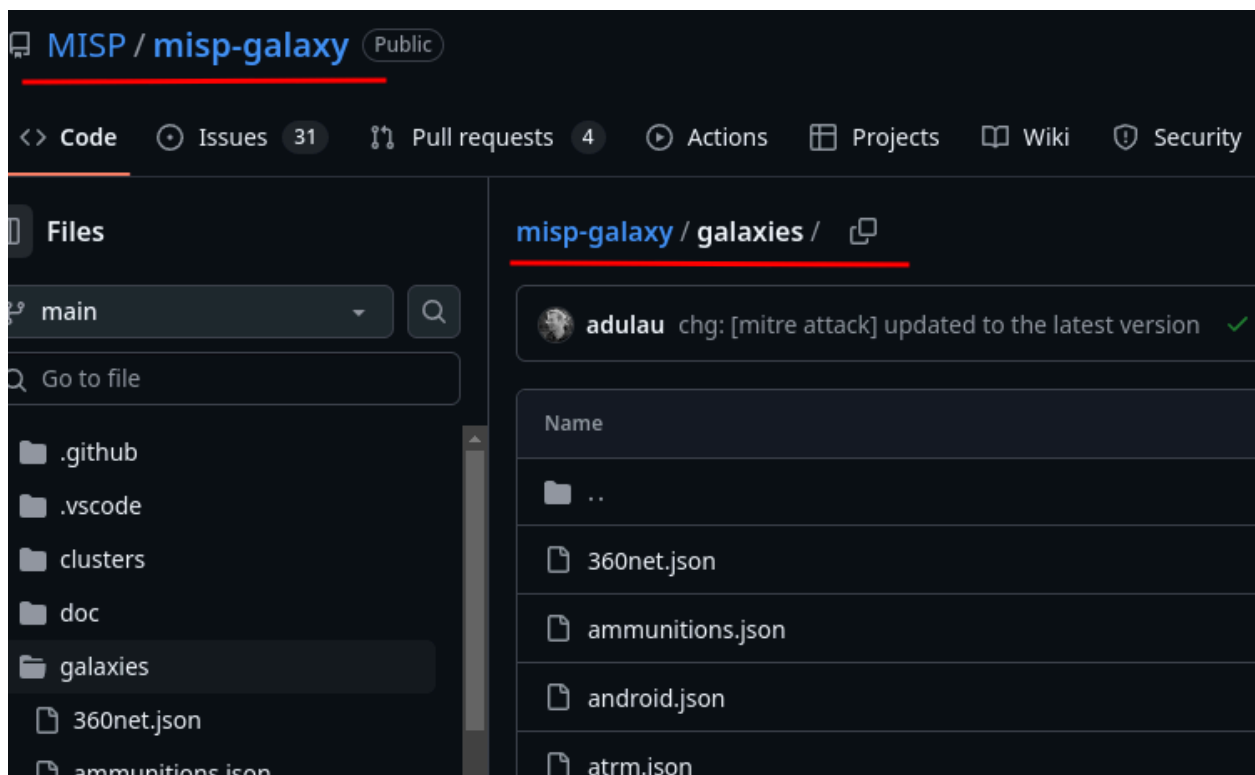
Define the maximum number of requests and the associated unit if applicable.

* Required field

Finalmente, podremos comprobar que se ha activado correctamente cuando veamos las siguientes métricas al lado del nombre del mismo.

| Max TLP | Max PAP | Rate Limit |
|--|-----------|------------|
| TLP:AMBER | PAP:AMBER | None |
| Edit Disable | | |

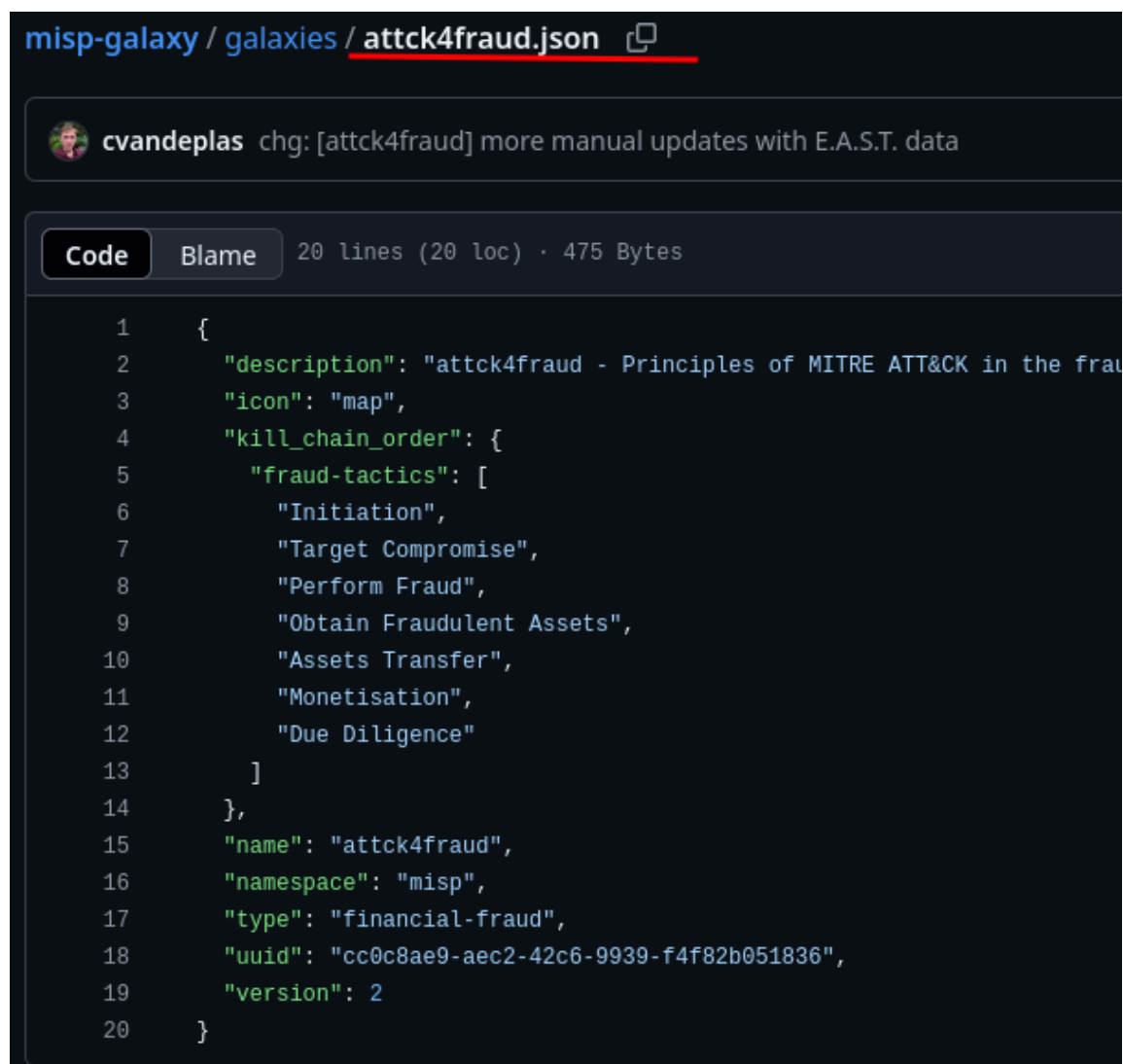
Ahora pasaremos a aprovisionar el contenedor Misp con una galaxia, para ello, nos dirigimos a su repositorio oficial y seleccionaremos la más adecuada a nuestro caso.



Una galaxia es un conjunto estructurado de información de inteligencia de amenazas, que agrupa técnicas, tácticas, actores de amenazas, campañas y otros conceptos en una forma estandarizada.

Por así decirlo, es una base de datos en la cual existen múltiples casos de ataque, estas se usan para determinar si una información recibida ha sido anteriormente detectada como ataque.

Se componen de datos “**JSON**”, el cual tendremos que copiar para implementarlo en nuestra instancia.



The screenshot shows the MISP-Galaxy interface for the 'attck4fraud' galaxy. The breadcrumb navigation at the top reads 'misp-galaxy / galaxies / attck4fraud.json'. Below this, a commit message from 'cvandeplas' is visible: 'chg: [attck4fraud] more manual updates with E.A.S.T. data'. The interface has tabs for 'Code' and 'Blame', with 'Code' selected. It indicates the file is 20 lines (20 loc) and 475 Bytes. The JSON content is displayed in a dark-themed editor with line numbers 1 through 20. The JSON object defines the 'attck4fraud' galaxy within the 'misp' namespace, categorized as 'financial-fraud'. It includes a description, an icon, a kill chain order, and a version number of 2.

```
1  {
2    "description": "attck4fraud - Principles of MITRE ATT&CK in the frau
3    "icon": "map",
4    "kill_chain_order": {
5      "fraud-tactics": [
6        "Initiation",
7        "Target Compromise",
8        "Perform Fraud",
9        "Obtain Fraudulent Assets",
10       "Assets Transfer",
11       "Monetisation",
12       "Due Diligence"
13     ]
14   },
15   "name": "attck4fraud",
16   "namespace": "misp",
17   "type": "financial-fraud",
18   "uuid": "cc0c8ae9-aec2-42c6-9939-f4f82b051836",
19   "version": 2
20 }
```

Posteriormente, en el apartado de **“Galaxies”** en la instancia de Misp, seleccionaremos **“Import Galaxy Cluster”** y pegaremos el **“JSON”** copiado anteriormente.

Home

Event Actions

Dashboard

Galaxies

Input Filters

Global Actions

Administration

Logs

API

List Galaxies

List Relationships

Import Galaxy Clusters

Import galaxy clusters

Paste a JSON of cluster to import or provide a JSON file below.

Warning: Use galaxy files generated by a MISP instance. Do NOT try to import a galaxy from the MISP-galaxy file

JSON

```
{
  "description": "attck4fraud - Principles of MITRE ATT&CK in the fraud domain",
  "icon": "map",
  "kill_chain_order": {
    "fraud-tactics": [
      "Initiation",
      "Target Compromise",
      "Perform Fraud",
      "Obtain Fraudulent Assets",
      "Assets Transfer",
      "Monetisation",
      "Due Diligence"
    ]
  },
  "name": "attck4fraud",
  "namespace": "misp",
  "type": "financial-fraud",
}
```

JSON file

Seleccionar archivo

Ningún archivo seleccionado

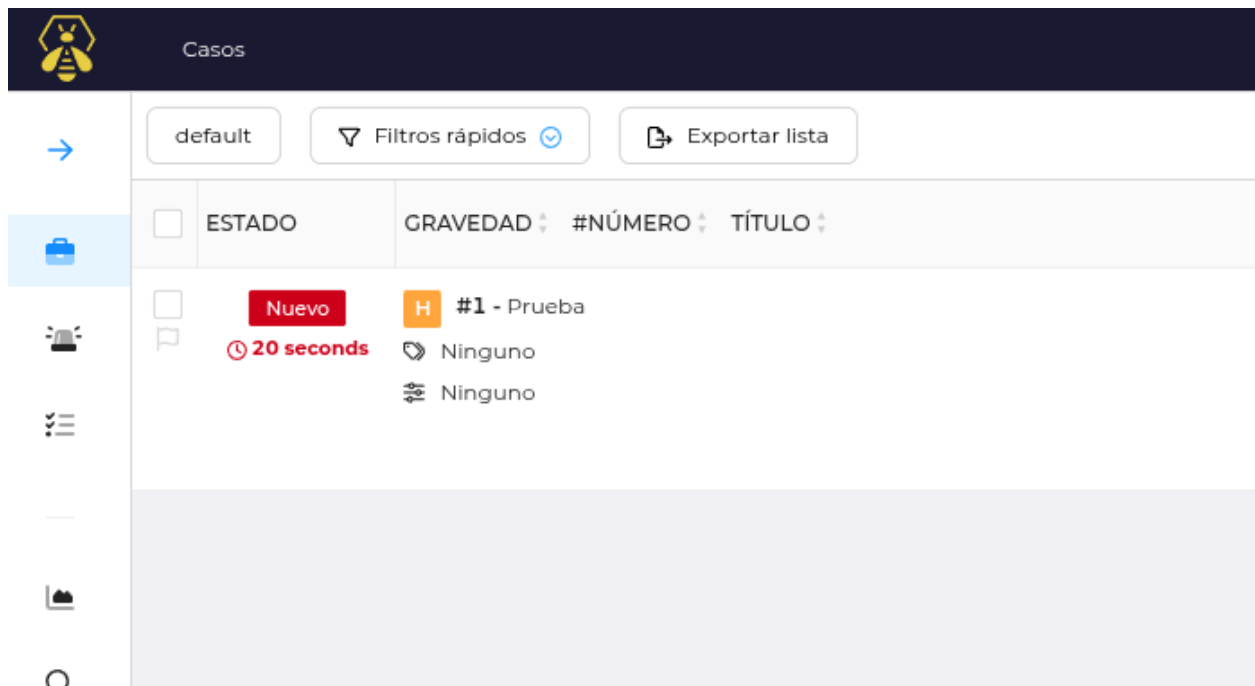
Submit

Visualización de Workflow en funcionamiento

Ahora vamos a pasar a visualizar el comportamiento de nuestro sistema, para ello, primero tendremos que iniciar el contenedor encargado de enviar información a Splunk.

```
pergo@Debian-Pergo:~/Escritorio/Sistema_SOC_Automatizado$ docker-compose -f test-container.yml up -d --build
WARNING: Found orphan containers (sistema_soc_automatizado_minio_1, sistema_soc_automatizado_cassandra_1, sistema_soc_automatizado_redis_1, sistema_soc_automatizado_cortex.local_1, sistema_soc_automatizado_splunk_1, sistema_soc_automatizado_misp.local_1) for this project. If you removed or renamed this service in your compose file, you can run this command: docker-compose rm --force --volumes sistema_soc_automatizado_minio_1, sistema_soc_automatizado_cassandra_1, sistema_soc_automatizado_redis_1, sistema_soc_automatizado_cortex.local_1, sistema_soc_automatizado_splunk_1, sistema_soc_automatizado_misp.local_1
Building sender
```

Inmediatamente, vemos que Splunk a creado un nuevo caso en TheHive, de esta forma, el equipo de Ciberseguridad de la organización podrá saber que se ha detectado un posible ataque.



The screenshot shows the TheHive web interface. At the top, there's a dark blue header with a yellow beehive icon and the word "Casos". Below the header, there's a navigation bar with a blue arrow icon, a "default" button, a "Filtros rápidos" button with a dropdown arrow, and an "Exportar lista" button. The main content area displays a table of cases. The first row is highlighted in light blue and contains the following information: a checkbox, the word "ESTADO", a red "Nuevo" label, an orange "H" icon, and the text "#1 - Prueba". Below this, there's a red clock icon with "20 seconds" and two "Ninguno" labels. The table has columns for "ESTADO", "GRAVEDAD", "#NÚMERO", and "TÍTULO".

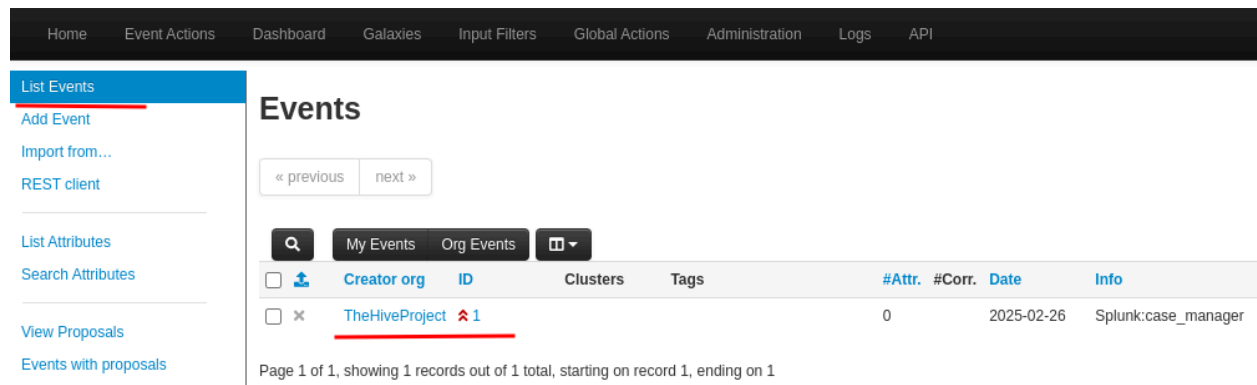
| | ESTADO | GRAVEDAD | #NÚMERO | TÍTULO |
|--------------------------|------------|----------|-------------|--------|
| <input type="checkbox"/> | Nuevo | H | #1 - Prueba | |
| <input type="checkbox"/> | 20 seconds | Ninguno | Ninguno | |

Inmediatamente, comprobamos que Cortex tiene un nuevo trabajo, esto quiere decir que se ha disparado el analizador, si este determina que es un ataque, se disparara el “responder” que bloqueará la dirección de la página que hemos simulado.

Desde Splunk, también podremos ver en tiempo real que está pasando, por ejemplo, aquí se ven las comunicaciones que está teniendo con el contenedor Misp.

| misp_instance ↕ | | |
|--|----------------------------|---|
| misp0 | | |
| Recent logs on MISP alert creation (last 60 minutes) | | |
| i | Time | Event |
| > | 2/26/25 10:59:55.012 AM | 2025-02-26 10:59:55,012 log_level=INFO pid=23856 tid=MainThread file=cim_actions.py nt" search_name="Test_Container_Actions" sid="rt_scheduler__admin__search__RMD5568: host = 1c19dfacadda source = /opt/splunk/var/log/splunk/misp_alert_create_event_modale |
| > | 2/26/25 10:59:55.011 AM | 2025-02-26 10:59:55,011 log_level=ERROR pid=23856 tid=MainThread file=cim_actions.py ponse={'_time': 1740567595.0111814, '_raw': "[MC503] DEBUG urllib3 POST request fail te_event" search_name="Test_Container_Actions" sid="rt_scheduler__admin__search__R host = 1c19dfacadda source = /opt/splunk/var/log/splunk/misp_alert_create_event_modale |
| > | 2/26/25 10:59:55.010 AM | 2025-02-26 10:59:55,010 log_level=ERROR pid=23856 tid=MainThread file=cim_actions.py ction_name="misp_alert_create_event" search_name="Test_Container_Actions" sid="rt_s e" host = 1c19dfacadda source = /opt/splunk/var/log/splunk/misp_alert_create_event_modale |
| > | 2/26/25 10:59:54.781 AM | 2025-02-26 10:59:54,781 log_level=INFO pid=23856 tid=MainThread file=cim_actions.py st_Container_Actions" sid="rt_scheduler__admin__search__RMD5568900294528c44b_at_174 host = 1c19dfacadda source = /opt/splunk/var/log/splunk/misp_alert_create_event_modale |
| > | 2/26/25 10:59:54.780 AM | 2025-02-26 10:59:54,780 log_level=INFO pid=23856 tid=MainThread file=cim_actions.py ctions" sid="rt_scheduler__admin__search__RMD5568900294528c44b_at_1740562274_2" ric host = 1c19dfacadda source = /opt/splunk/var/log/splunk/misp_alert_create_event_modale |

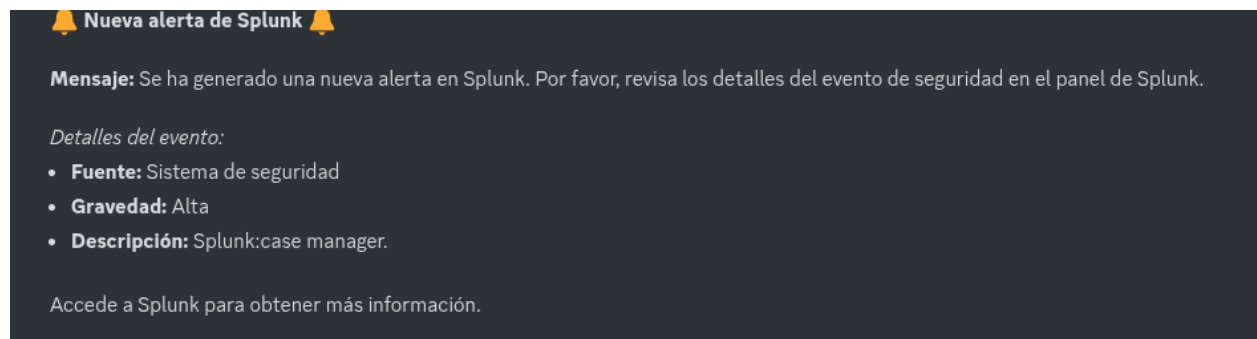
En Misp, Splunk creará un nuevo caso, esta información será comparada con la galaxia previamente importada para ver si anteriormente se ha detectado un ataque similar, si es el caso, se hará saber a través del panel de Misp.



The screenshot shows the Misp interface with a top navigation bar and a left sidebar. The main content area is titled "Events" and contains a table of event records. The table has columns for Creator org, ID, Clusters, Tags, #Attr., #Corr., Date, and Info. A single record is displayed for "TheHiveProject" with ID 1, 0 attributes, 1 correction, and a date of 2025-02-26. The info field shows "Splunk:case_manager".

| Creator org | ID | Clusters | Tags | #Attr. | #Corr. | Date | Info |
|----------------|----|----------|------|--------|--------|------------|---------------------|
| TheHiveProject | 1 | | | 0 | 1 | 2025-02-26 | Splunk:case_manager |

En discord, se ha creado una nueva alerta con éxito, por lo que el administrador del sistema también podría estar al tanto de lo que está pasando en el mismo.



The screenshot shows a Discord message with a yellow bell icon and the text "Nueva alerta de Splunk". The message content is as follows:

Mensaje: Se ha generado una nueva alerta en Splunk. Por favor, revisa los detalles del evento de seguridad en el panel de Splunk.

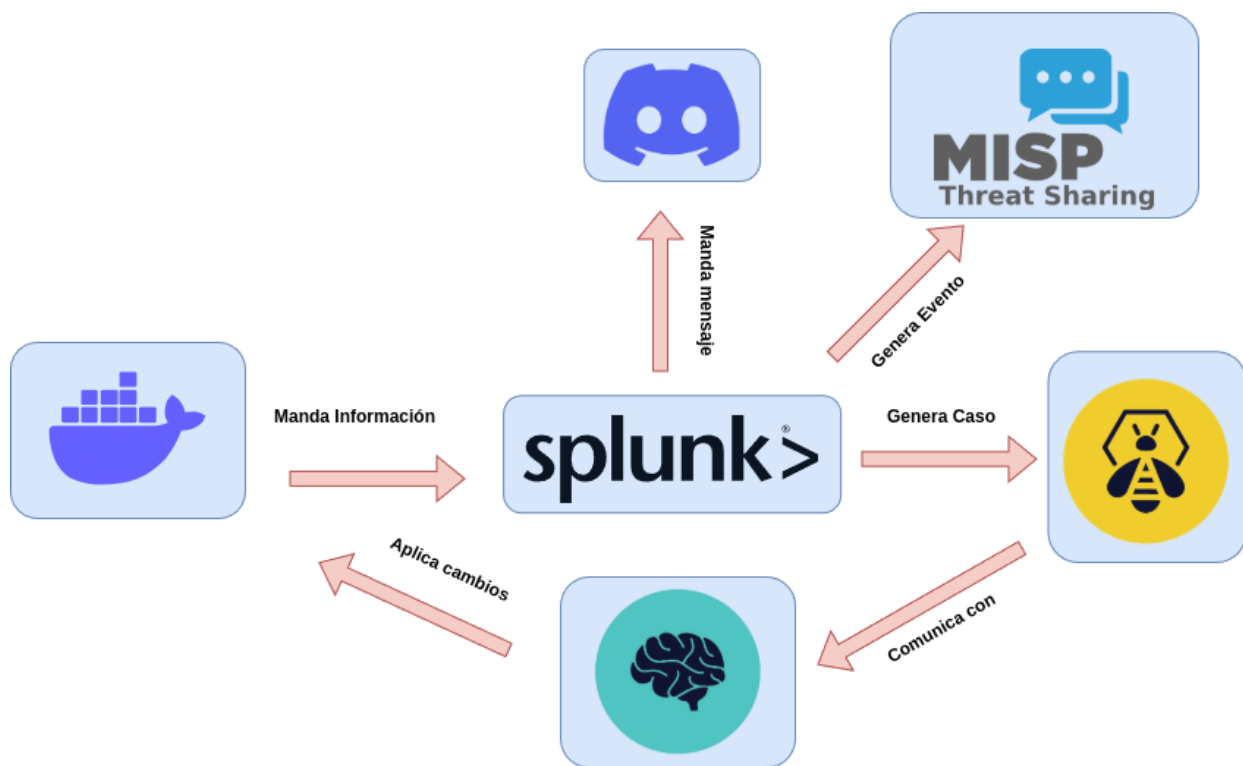
Detalles del evento:

- **Fuente:** Sistema de seguridad
- **Gravedad:** Alta
- **Descripción:** Splunk:case manager.

Accede a Splunk para obtener más información.

Conclusión y aclaraciones

Con esto, hemos logrado crear un sistema de monitorización, análisis y respuesta en tiempo real ante posibles ataques de seguridad, a continuación, se muestra el WorkFlow del sistema de forma gráfica.



Tras ver de forma gráfica el WorkFlow que tendrá el sistema configurado, pasaremos a las aclaraciones y a la conclusión final del proyecto.

Hay que tener en cuenta que esta solo es una configuración entre las muchas disponibles en el sistema, este se puede configurar para analizar y responder a una infinidad de ataques mediante las configuraciones específicas.

En cualquier caso, la integración entre las instancias será muy exacta a la desarrollada en ese proyecto.

A continuación, vamos a describir Workflow que ha sido creado en el desarrollo del proyecto.

1. Test-Container → Manda información a de prueba a Splunk de forma constante.

2. Splunk → Recoge, analiza, y dispara tigers en base a la información recibida, además proporciona dashboards útiles para la interpretación del flujo de datos en el sistema.

3. TheHive → Sirve como gestor de casos para que el equipo de la empresa sepa qué está pasando en el sistema, además de permitir crear casos personalizados para gestionar tareas rutinarias.

4. Cortex → Analiza la información de los casos creados en TheHive por Splunk y actúa de forma automática en base a las configuraciones que hemos implantado.

5. Misp → Compara la información recibida por Splunk con las galaxias importadas para determinar si anteriormente ha sido detectada como ataque.

Con esta configuración de prueba, hemos conseguido comprobar que la integración entre los servicios ha sido completamente exitosa y el sistema está preparado para ser configurado en base a las necesidades requeridas y ofrecer gestión, análisis y respuesta de forma automática ante posibles ataques.

Webgrafía

Inteligencias artificiales usadas para el proyecto.

[Chat GPT](#)

[Gemini.](#)

[Copilot.](#)

Páginas Webs usadas en el proyecto.

[Guia de ayuda integracion TheHive + Cortex +MISP.](#)

[Repositorio de ayuda integracion Splunk + TheHive + Cortex.](#)

[Repositorio de Galaxias de Misp.](#)

[Página usada para comprobar las URL.](#)

[Documentación oficial de Splunk.](#)

[Repositorio oficial TheHiveProject.](#)

Proveedores de servicios usados para el proyecto.

[Hetzner Cloud.](#)

[Nominalia.](#)

[OVH Cloud.](#)

[Google Drive](#)

Contacto con el creador del proyecto.

