

Anexo 2

Despliegue de Netskope CE en Red Hat 9

Jordi Guillem Ferrer Bozzano

Máster Universitario en Ciberseguridad y Privacidad
Seguridad empresarial

Iñaki Moreno Fernández

Víctor García Font

10 de enero de 2023

Despliegue de Netskope CE en Red Hat 9

1. INSTALACIÓN DE NETSKOPE CLOUD EXCHANGE	3
2. AÑADIR EL TENANT DE NETSKOPE.....	7
3. CONFIGURACIÓN DE LOG SHIPPER	8
4. REFERENCIAS.....	11

ÍNDICE DE ILUSTRACIONES

ILUSTRACIÓN 1: SELECCIÓN DE VERSIÓN EN LA INSTALACIÓN NETSKOPE CE	4
ILUSTRACIÓN 2: ESPECIFICAR NOMBRE TENANT EN LA INSTALACIÓN DE NETSKOPE CE.....	4
ILUSTRACIÓN 3: ESPECIFICAR PROTOCOLO Y PUERTO PARA EL NETSKOPE CE UI.....	4
ILUSTRACIÓN 4: INTRODUCIR CREDENCIALES TOKEN Y MONGODB PARA NETSKOPE CE	5
ILUSTRACIÓN 5: EJECUTAR INSTALACIÓN DE NETSKOPE CE	5
ILUSTRACIÓN 6: VENTANA INICIAL DE NETSKOPE CE UI	6
<i>ILUSTRACIÓN 7: AÑADIR TENANT EN NETSKOPE CE UI.....</i>	<i>7</i>
ILUSTRACIÓN 8: VISTA DE TENANT EN NETSKOPE CE UI	7
ILUSTRACIÓN 9: ACTIVACIÓN DE MÓDULO LOG SHIPPER EN NETSKOPE CE UI	8
ILUSTRACIÓN 10: SELECCIÓN DE PLUGINS PARA LOG SHIPPER.....	8
ILUSTRACIÓN 11: CONFIGURACIÓN DE PLUGIN DE NETSKOPE PARA LOG SHIPPER	9
<i>ILUSTRACIÓN 12: CONFIGURACIÓN DE PLUGIN DE IBM QRADAR PARA LOG SHIPPER.....</i>	<i>9</i>
ILUSTRACIÓN 13: CREACIÓN DE MAPPING DE LOG SHIPPER.....	10

1. Instalación de Netskope Cloud Exchange

La instalación de Netskope Cloud Exchange se realiza desde un sistema operativo Red Hat 9 actualizado con los últimos paquetes del repositorio oficial de Red Hat. Los pasos para poder instalar Netskope CE son los siguientes:

1.1. Instalar los requisitos correspondientes a Red Hat 9:

```
#> yum -y install podman podman-compose podman-plugins python3
```

1.2. Asegurarse que el idioma del sistema esté en inglés.

```
#> echo $LANG
```

En el caso que no esté en inglés ejecutar el siguiente comando:

```
#> localectl set-locales LANG=en_US
```

1.3. Clonar el proyecto GIT:

```
#> mkdir /opt/netskope
```

```
#> git clone https://github.com/netskopeoss/ta\_cloud\_exchange
```

1.4. Ejecutar el comando de configuración de Netskope CE:

```
#> cd /opt/netskope/ta_cloud_exchange
```

```
#> python3 ./setup
```

1.5. Seleccionar la versión CE v4-latest:

1.8. Especificar la contraseña para el **JSON Web Secret** (JWT Secret) y el **Maintenance password** que se usará para Rabbit y MongoDB:

```
> Enter a JWT Secret which will be used for signing the authentication tokens:
> Enter maintenance password that will be used for RabbitMQ and MongoDB services (This password can be set only once):
> Confirm maintenance password:
> Do you want to enable TLSv1.2 along with TLSv1.3 for CE UI (Default: "No"):

Setup completed successfully...

Execute this command to start CE:
  > sudo ./start

Please re-run the setup script to update any parameter.

Warning: It is recommended to take an external backup of the .env file located in this directory.
[root@BTPNETSCOPECE01 ta cloud exchange]#
```

Ilustración 4: Introducir credenciales token y MongoDB para Netskope CE

1.9. Ejecutar comando de instalación con `./start`, la salida es la siguiente:

```
er-number=1 --label com.docker.compose.service=mongodb-primary -e MONGODB_ADVERTISED_HOSTNAME=mongodb-primary -e MONGODB_ROOT_PASSWORD=8gZ4KjLq29Qtyuie -e MONGODB_USERNAME=cteadmin -e MONGODB_PASSWORD=8gZ4KjLq29Qtyuie -e MONGODB_DATABASE=cte -e HTTP_PROXY= -e HTTPS_PROXY= -v /opt/netskope/ta_cloud_exchange/data/mongodb:/bitnami/mongodb:z --net ta_cloud_exchange_default --network-alias mongodb-primary --log-driver=k8s-file --log-opt=max-size=10m --log-opt=max-file=5 --restart always index.docker.io/bitnami/mongodb:4.4f67b62db20a2d8f5a8a5bc819c772af31502ac75a483fe6a6abd7ea85b36d021
exit code: 0
[['podman', 'network', 'exists', 'ta_cloud_exchange_default']]
podman run --name=ta_cloud_exchange_core_1 -d --label com.centurylinklabs.watchtower.enable=true --label io.podman.compose.config-hash=123 --label io.podman.compose.project=ta_cloud_exchange --label io.podman.compose.version=0.0.1 --label com.docker.compose.project=ta_cloud_exchange --label com.docker.compose.project.working_dir=/opt/netskope/ta_cloud_exchange --label com.docker.compose.project.config_files=podman-compose.yml --label com.docker.compose.container-number=1 --label com.docker.compose.service=core -e MONGO_CONNECTION_STRING=mongodb://cteadmin:8gZ4KjLq29Qtyuie@mongodb-primary:27017/cte -e RABBITMQ_CONNECTION_STRING=amqp://user:8gZ4KjLq29Qtyuie@rabbitmq-stats -e JWT_SECRET=ejV2PPYP0S445hyu -e JWT_ALGORITHM=HS256 -e ENABLE_CELERY_BEAT=true -e WATCHTOWER_HTTP_API_TOKEN=token -e ANALYTICS_BASE_URL=https://reporting.netskope.tech -e ANALYTICS_TOKEN=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpbnN0YWxsYXRpb25faWQiOiJjMDAyIn0.w8SVrTcDjk8PkR4IcbWgWoyf6-0WfCUyOoCTgZvqHgc -e MAX_MAINTENANCE_WINDOW_MINUTES=15 -e PULL_THREADS=6 -e MAX_WAIT_ON_LOCK_IN_MINUTES=240 -e HTTP_PROXY= -e HTTPS_PROXY= -e BETA_OPT_IN=No -v /opt/netskope/ta_cloud_exchange/data/custom_plugins:/opt/netskope/plugins/custom_plugins:z -v /run/docker.sock:/var/run/docker.sock --net ta_cloud_exchange_default --network-alias core --log-driver=k8s-file --log-opt=max-size=10m --log-opt=max-file=5 --restart always index.docker.io/netskopetechnicalalliances/cloudexchange:core4-latest
61cdc09a306e4f2d20e8bb50257060ef427d681056e0f1755586625e8a941bcf
exit code: 0
[['podman', 'network', 'exists', 'ta_cloud_exchange_default']]
podman run --name=ta_cloud_exchange_ui_1 -d --label com.centurylinklabs.watchtower.enable=true --label io.podman.compose.config-hash=123 --label io.podman.compose.project=ta_cloud_exchange --label io.podman.compose.version=0.0.1 --label com.docker.compose.project=ta_cloud_exchange --label com.docker.compose.project.working_dir=/opt/netskope/ta_cloud_exchange --label com.docker.compose.project.config_files=podman-compose.yml --label com.docker.compose.container-number=1 --label com.docker.compose.service=ui -e CE_API_URL=http://core -e TLS_VERSION=TLSv1.3 -v /opt/netskope/ta_cloud_exchange/data/ssl_certs:/tmp/ssl_certs:z --net ta_cloud_exchange_default --network-alias ui --log-driver=k8s-file --log-opt=max-size=10m --log-opt=max-file=5 -p 443:3000 --restart always index.docker.io/netskopetechnicalalliances/cloudexchange:ui4-latest
fdbccd5373d95a239911988bb6c6140360ec4ccc6198c27eab8dc679cf75a0a1
exit code: 0
[root@BTPNETSCOPECE01 ta cloud exchange]#
```

Ilustración 5: Ejecutar instalación de Netskope CE

Todos los *exit codes* deberán estar en 0 para saber si la instalación ha sido satisfactoria.

1.10. Comprobar el acceso a Netskope CE UI mediante la URL correspondiente <https://btpnetskopece01.tous.net>

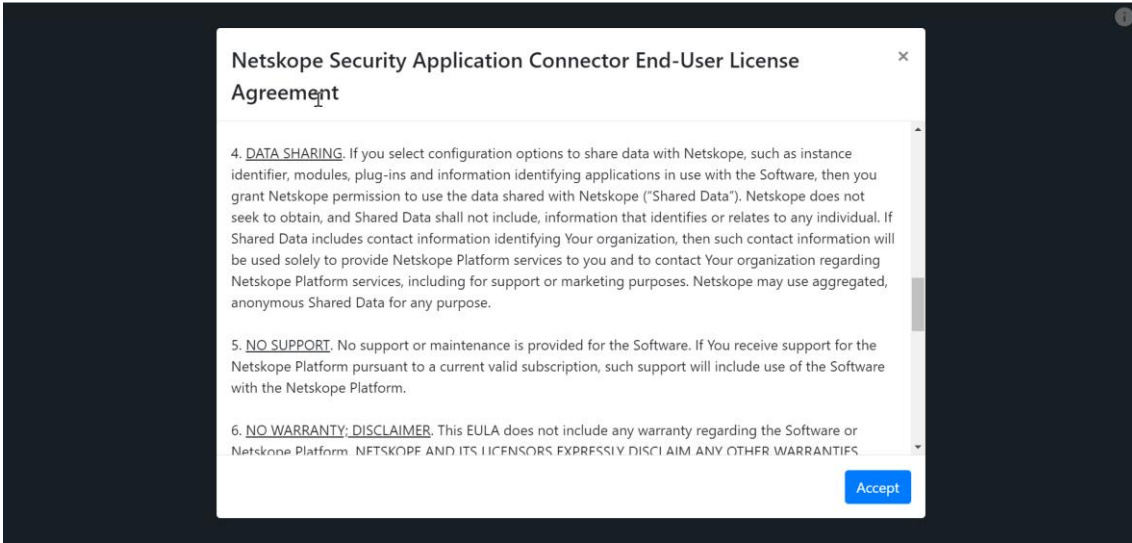


Ilustración 6: Ventana inicial de Netskope CE UI

2. Añadir el tenant de Netskope

- 2.1. Es necesario asociar el Netskope Cloud Exchange con el tenant de Netskope, para ello hay que obtener los API v1 y v2 del tenant e introducirlos dentro de los campos de creación en **Settings > Netskope tenants > Add Tenant** (En la imagen no se muestra la información de API Token)

Add Tenant ×

Name ⓘ
Zanomme

Tenant Name ⓘ
zanomme.eu

V1 API Token ⓘ
Enter API token

V2 API Token (Optional) ⓘ
Enter v2 API token

Use Iterator Endpoint ⓘ

Initial Range (in days) ⓘ
Initial Range (in days)

Use System Proxy ⓘ

Ilustración 7: Añadir tenant en Netskope CE UI

- 2.2. Comprobar la creación del *tenant* como se muestra en la imagen siguiente:

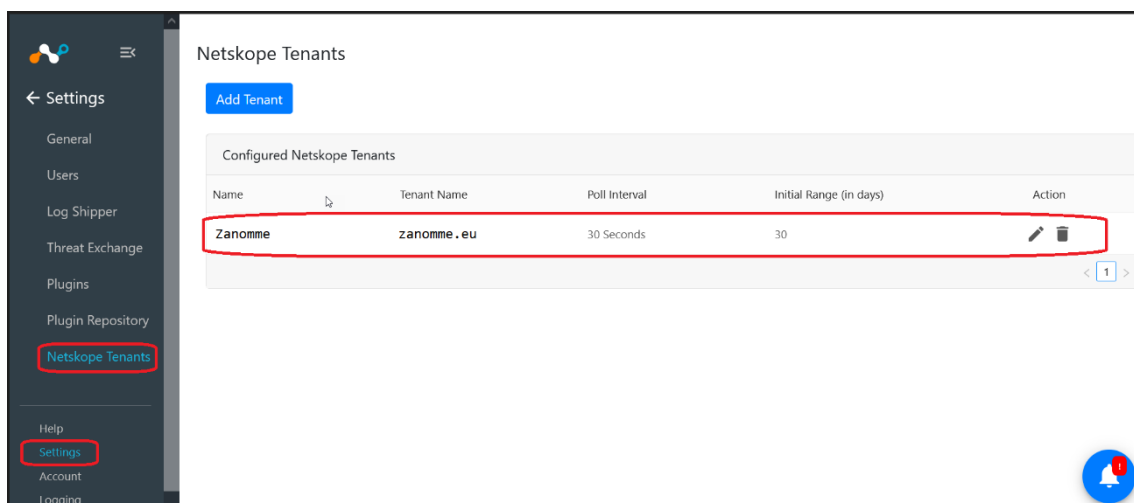


Ilustración 8: Vista de tenant en Netskope CE UI

3. Configuración de Log Shipper

3.1. Activar el módulo de Log Shipper en **Settings > General > Log Shipper**:

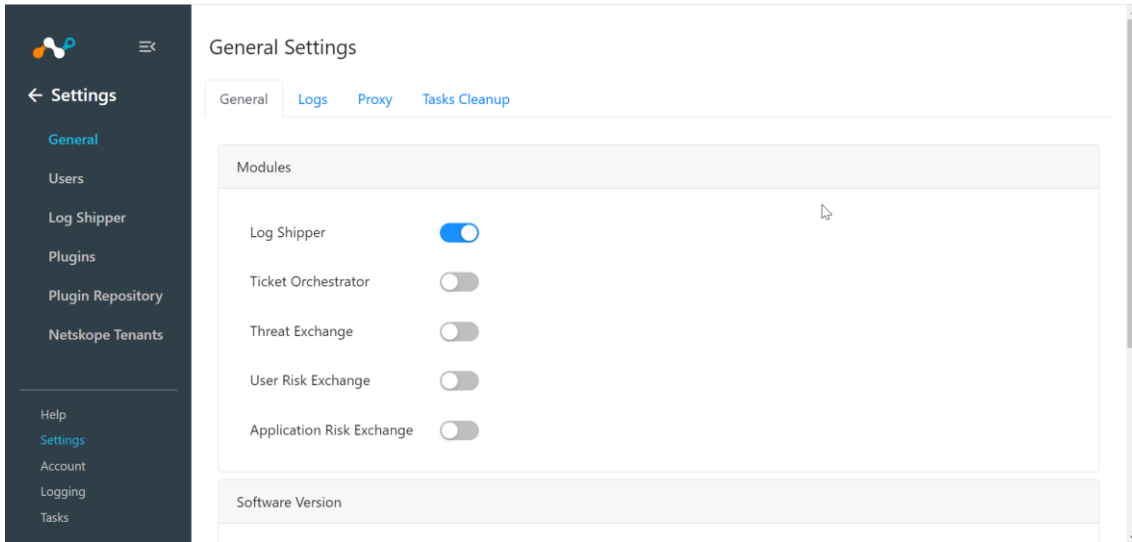


Ilustración 9: Activación de módulo Log Shipper en Netskope CE UI

3.2. Seleccionar los plugins necesarios desde **Settings > Plugins** en este caso se selecciona el correspondiente de **Netskope** y el de **IBM Security QRadar**:

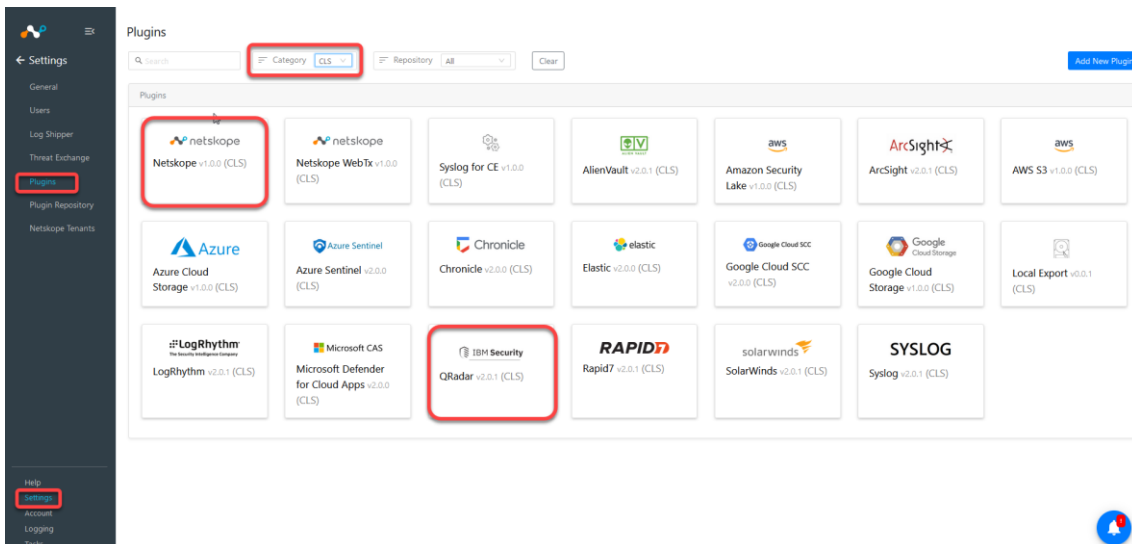


Ilustración 10: Selección de plugins para Log Shipper

3.3. Configurar el correspondiente a Netskope:

Netskope v1.0.0

This plugin is used to fetch alerts and events from Netskope.

- Basic Information
- 2 Configuration Parameters**

Alert Type ?

Anomaly x Compromised Credential x Policy x Legal Hold x Malsite x Malware x DLP x Security Assessment x
Quarantine x Remediation x

Event Type ?

Page x Application x Audit x Infrastructure x Network x

Initial Range (in hours) ?

1

Previous

Ilustración 11: Configuración de plugin de Netskope para Log Shipper

- 3.4.** Configurar el plugin de IBM Security QRadar, en el caso del laboratorio realizado en este proyecto se especifica la IP del colector interno el cual acepta conexiones Syslog en la red interna, este colector interno enviará los eventos al Cloud de IBM Qradar:

QRadar v2.0.1

This plugin is used to ingest data to QRadar platform. To access the plugin, you would need the credentials of QRadar platform.

- Basic Information
- 2 Configuration Parameters**

QRadar Server ?
192.168.170.100

QRadar Format ?
CEF

QRadar Protocol ?
TCP

QRadar Port ?
514

QRadar Certificate ?
QRadar Certificate

Log Source Identifier ?
zanomnenskope

Ilustración 12: Configuración de plugin de IBM Qradar para Log Shipper

- 3.5. Desde **Log Shipper > SIEM Mappings** crear el *mapping* correspondiente al origen **zanomme** (Netskope Plugin) y como destino al **SIEMSOC**.

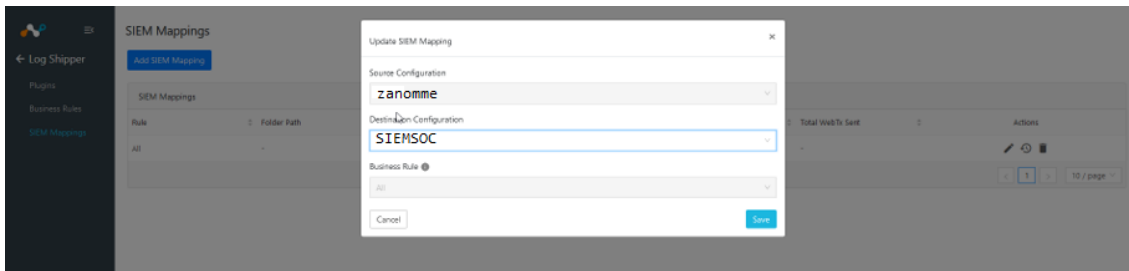


Ilustración 13: Creación de mapping de Log Shipper

- 3.6. Una vez guardado el *mapping* los eventos del tenant de Netskope se deberían de empezar a recibir en el colector interno de IBM QRadar para poder enviarlo definitivamente al Cloud SIEM de QRadar.

4. Referencias

Netskope Cloud Exchange. (s. f.). <https://docs.netskope.com/en/netskope-cloud-exchange.html>