10.5 SECURITY

La última de las funcionalidades y quizás la más solicitada e importante es la securización del Stack ELK. Esto incluye desde cifrado de comunicaciones entre los nodos, flujo de datos desde las fuentes cifrado, acceso mediante usuario y contraseña a Kibana, creación de distintos roles con sus correspondientes permisos... etc.

En este tema veremos la configuración de cada uno de los componentes, generación de certificados y cada uno de los pasos necesarios para convertir a nuestro cluster en una arquitectura cifrada, con accesos y más segura.

Activar el uso de security

El primer paso necesario será activar el uso de este plugin desde la configuración de elasticsearch `elasticsearch.yml` añadiendo la siguiente linea:

```
xpack.security.enabled: true
```

Cifrado de comunicaciones

Si escucháramos el tráfico que llega de eventos o simplemente las comunicaciones entre los distintos nodos se puede comprobar que viajan en texto claro. Un gran peligro ante un ataque de `man-in-the-middle` que podría escuchar todo el tráfico y las comunicaciones.

Los pasos serán los siguientes:

- **Generación de los certificados de los nodos**

TLS requiere certificados X.509 para llevar a cabo el cifrado y autenticación de las aplicaciones con las que se comunica, y para que las comunicaciones entre los nodos sean seguras, dichos certificados han debido ser validados. Para ello se requiere que una Autoridad Certificadora (CA) en la que se confía firme los certificados.

Podemos crear un directorio con los permisos adecuados para tener todos los certificados dentro:

```
mkdir /etc/elasticsearch/certs
cd /etc/elasticsearch/certs
```

En el caso de que tengamos la posibilidad de crear los certificados a través de nuestra compañía estupendo, en caso contrario ELK proporciona una herramienta para facilitar este proceso llamdada `elasticsearch-certutil` . Esta herramienta genera una CA y firma los certificados para los nodos.

```
/usr/share/elasticsearch/bin/elasticsearch-certutil ca
```

Ahora será necesario generar un certificado y clave privada para cada uno de los nodos:

```
/usr/share/elasticsearch/bin/elasticsearch-certutil cert --ca elastic-stack-ca.p12
 --name elastic01 --ip 192.168.1.31
/usr/share/elasticsearch/bin/elasticsearch-certutil cert --ca elastic-stack-ca.p12
 --name elastic02 --ip 192.168.1.32
/usr/share/elasticsearch/bin/elasticsearch-certutil cert --ca elastic-stack-ca.p12
 --name elastic03 --ip 192.168.1.33
```

La salida será un keystore PKCS#12 que incluye el certificado y la clave del nodo y el certificado de la CA.

Una vez generados, deberemos crear en el resto de nodos el mismo directorio y distribuir sus correspondientes certificados mediante `scp` y posteriormente modificar en *todos* ellos los permisos de estos directorios para que elasticsearch tenga acceso a los mismos.

En mi caso al haber ejecutado estos comandos en `elastic01` solo tendré que pasar los archivos al segundo y tercer nodo.

```
scp elastic02.p12 root@192.168.1.32:/etc/elasticsearch/certs
scp elastic03.p12 root@192.168.1.33:/etc/elasticsearch/certs
```

Y modificamos los permisos de estos directorios:

```
chown -R root:elasticsearch certs
chmod 660 certs/*
```

- **Cifrar la comunicaciones entre los nodos**

Una vez estos certificados han sido creados, deberá añadirse en `elasticsearch.yml` de cada uno de los nodos los parámetros necesarios para que cifren las comunicaciones haciendo uso de estos certificados recien creados. Hay que recordar cambiar en las siguientes líneas el nombre del certificado dependiendo del nodo en el que se esté configurando:

```
xpack.security.transport.ssl.enabled: true
xpack.security.transport.ssl.verification_mode: certificate
xpack.security.transport.ssl.keystore.path: /etc/elasticsearch/certs/elastic01.p12
xpack.security.transport.ssl.truststore.path: /etc/elasticsearch/certs/elastic01.p
12
```

En el caso de haber añadido una contraseña a los certificados, deberá añadirse al keystore de Elasticsearch:

```
bin/elasticsearch-keystore add xpack.security.transport.ssl.keystore.secure_passwo
rd
bin/elasticsearch-keystore add xpack.security.transport.ssl.truststore.secure_pass
word
```

Y ya se puede reiniciar el clúster para comprobar que los cambios son correctos y están funcionando.

```
service elasticsearch restart
```

- **Cifrado de las comunicaciones HTTP**

Aprovechando que los certificados ya se encuentran creados, se pueden cifrar también toda comunicación de agentes externos con el clúster por HTTP.

```
xpack.security.http.ssl.enabled: true
xpack.security.http.ssl.keystore.path: /etc/elasticsearch/certs/elastic01.p12
xpack.security.http.ssl.truststore.path: /etc/elasticsearch/certs/elastic01.p12
```

Y se puede reiniciar elasticsearch para que los cambios surtan efecto:

```
service elasticsearch restart
```

- **Configuración de Kibana**

El primer paso dentro de Kibana será modificar las contraseñas de acceso de los distintos usuarios que vienen por defecto. Para ello ELK ofrece una herramienta que nos facilite este trabajo `elasticsearch-setup-passwords`:

```
/usr/share/elasticsearch/bin/elasticsearch-setup-passwords interactive
```

Una vez modificadas las contraseñas, deberemos cambiar la configuración de Kibana para que pueda conectarse de nuevo al clúster añadiendo el usuario y contraseña establecidos en el paso anterior, utilizando https y activando el `encriptionKey` dentro de `kibana.yml`:

```
elasticsearch.url: "https://192.168.1.31:9200"
elasticsearch.username: "kibana"
elasticsearch.password: "toor00"
xpack.security.encryptionKey: "something_at_least_32_characters"
```

Y el último parámetro a configurar será la CA en formato PEM ya que hemos sido nosotros los que la hemos generado, Kibana deberá confiar también en ella para conectarse. El problema es que con la herramienta se ha generado en formato `.p12` y es necesario que sea en formato `pem` por lo que para sacar uno del otro se hará de la siguiente forma:

```
cd /etc/elasticsearch/certs
openssl pkcs12 -in elastic-stack-ca.p12 -clcerts -nokeys -out elastic-stack-ca.pem
mkdir /etc/kibana/certs
mv elastic-stack-ca.pem /etc/kibana/certs/
cd /etc/kibana/certs/
chmod 664 elastic-stack-ca.pem
chown root:kibana elastic-stack-ca.pem
```

Y añadimos en Kibana este `pem` generado de la CA.

```
elasticsearch.ssl.certificateAuthorities: /etc/kibana/certs/elastic-stack-ca.pem
```

Y ya se puede reiniciar Kibana:

```
service kibana restart
```

En el caso de que dispongamos de los certificados `crt` y `key` para la máquina de Kibana, también se podrán añadir los siguientes parámetros para que Kibana quede publicada por HTTPS en vez de HTTP.

```
server.ssl.enabled: true
server.ssl.key: /path/to/your/server.key
server.ssl.certificate: /path/to/your/server.crt
```

Y con estas configuraciones se da por finalizada la securización del clúster, y se puede proceder a la creación de roles y usuarios.