

ELK



Creación de un SIEM ELK

A lo largo de este trabajo crearemos un sistema gestor de eventos (SIEM) mediante el uso de un (ELK), es decir usaremos “Elasticsearch” y “Kibana” como base de datos y visualizador de eventos.

Estos eventos serán creados por las herramientas “Suricata” y “Rsyslogs” y recogidos por las herramientas “Filebeat” y “Logstash”.

C I B E R S E G U R I D A D
I N C I D E N T E S D E
C I B E R S E G U R I D A D
E R I C S U A R E Z V A Z Q U E Z
0 1 / 0 4 / 2 0 2 4

La combinación exitosa de todas estas herramientas será el sistema gestor de eventos.

INDICE

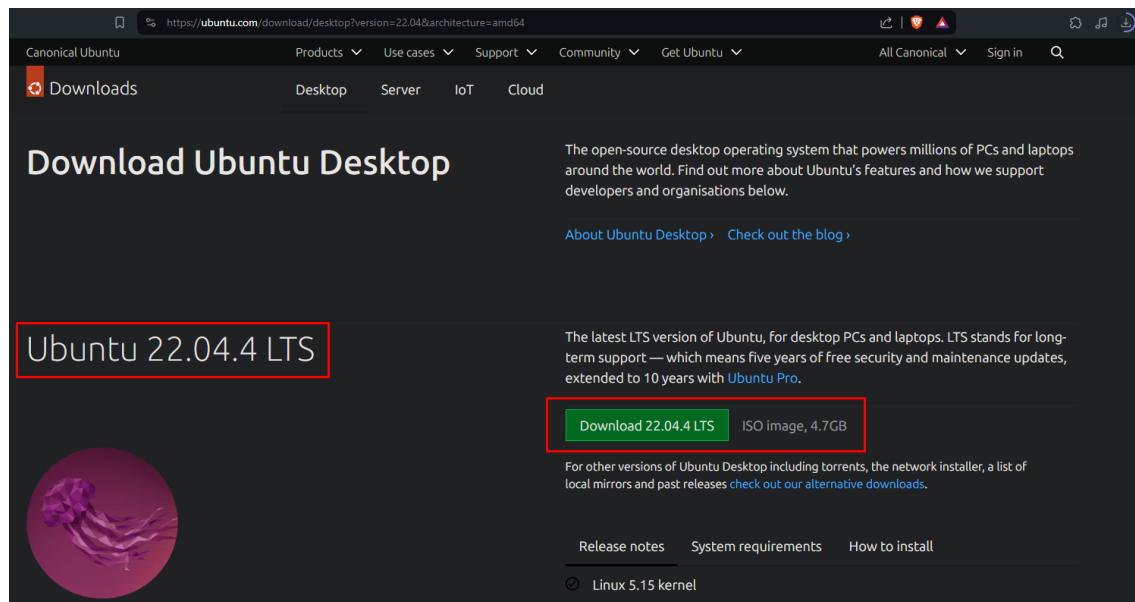
Creación de máquina virtual.....	2
Herramientas.....	3
Suricata.....	3
Rsyslog	3
FileBeat	4
ElasticSearch	4
Logstash.....	5
Kibana	5
Objetivo principal del trabajo	6
¿Cómo lograremos esto?	6
Instalaciones de Herramientas Necesarias	7
Suricata.....	7
Rsyslog	8
Filebeat	8
ElasticSearch	9
Logstash.....	9
Kibana	10
Configuración de Herramientas.....	11
Configuración y reglas personales de Suricata.....	11
Configuración de Rsyslog	14
Configuración de ElasticSearch.....	16
Configuración de Kibana.....	17
Página de ElasticSearch	18
Configuración de Logstash.....	21
Configuración de FileBeat.....	22
Página ElasticSearch	26
Securización del ELK	32
Plugin	32
Credenciales de inicio	33
HTTPS con ElasticSearch	35
Final	38

Creación de máquina virtual

La máquina en la que elaboraremos el sistema gestor de eventos será una máquina Ubuntu 22.04. La podemos descargar desde el siguiente enlace:

<https://ubuntu.com/download/desktop?version=22.04&architecture=amd64>

Esta es una de las distribuciones de **Ubuntu** más actuales lo cual nos ayudará en la instalación de nuestro **SIEM**.



Esta imagen la montaremos en una máquina virtual de VMWare y, en ella, actualizaremos los paquetes necesarios e instalaremos todas las herramientas. Antes de empezar con las instalaciones de los programas, vamos a conocer el papel de estas en el sistema gestor.

Herramientas

Suricata

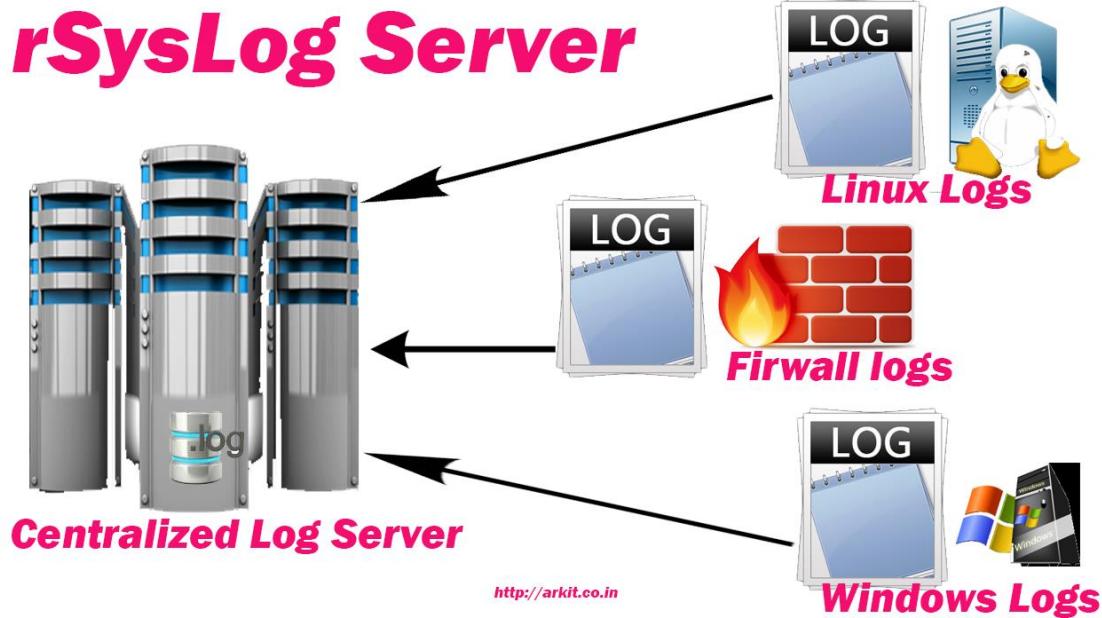
Suricata es una herramienta esencial para **monitorear y proteger** las redes de cualquier tipo de actividad maliciosa o no autorizada, proporcionando a los **administradores** de sistemas y equipos de **seguridad** la capacidad de **detectar, investigar y responder** a las amenazas en tiempo real.



Rsyslog

Rsyslog es una herramienta que ayuda a la gestión de registros de eventos en sistemas Linux, proporcionando capacidades de **recolección, procesamiento, almacenamiento y envío** de registros de manera eficiente y segura.

Es fundamental para la **monitorización, diagnóstico, análisis de seguridad, y cumplimiento** normativo en entornos de servidores y redes.



FileBeat

Filebeat es una herramienta **clave** para la **recopilación** y **envío de datos de registro** en entornos distribuidos, facilitando el **traspaso de datos** en **Elasticsearch** y **Logstash** para su posterior **análisis, búsqueda y visualización** utilizando herramientas como **Kibana**. Es especialmente útil para la **monitorización de sistemas, la detección de problemas y la investigación de incidentes** en entornos de producción.



ElasticSearch

Elasticsearch es una poderosa plataforma de **búsqueda y análisis de datos** que proporciona capacidades avanzadas de **búsqueda, análisis y almacenamiento de datos** en tiempo real. Es ampliamente utilizado como base de datos en una variedad de casos de uso, incluyendo la **búsqueda de texto completo, la analítica de registros de eventos, la monitorización de sistemas, la analítica de negocio**, y muchos más.



Logstash

Logstash es una herramienta utilizada para la **ingestión, transformación y enriquecimiento de datos en entornos de análisis de datos y procesamiento de registros de eventos**.

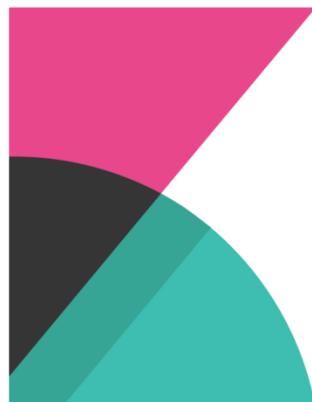
Proporciona capacidades avanzadas para **recopilar, procesar y enrutar datos** de manera eficiente y flexible, lo que lo hace adecuado para una amplia variedad de casos de uso, desde la **monitorización de sistemas hasta el análisis de datos** en tiempo real para posteriormente visualizar dichos datos con otra herramienta.



logstash

Kibana

Kibana es una herramienta principal para la **visualización y análisis de datos almacenados en Elasticsearch**, proporcionando capacidades avanzadas para **explorar, analizar y representar** visualmente datos de una manera significativa y comprensible. Es ampliamente utilizado en una variedad de casos de uso, incluyendo la **monitorización de sistemas, la analítica de negocio, la visualización de datos geoespaciales**, y muchos más.



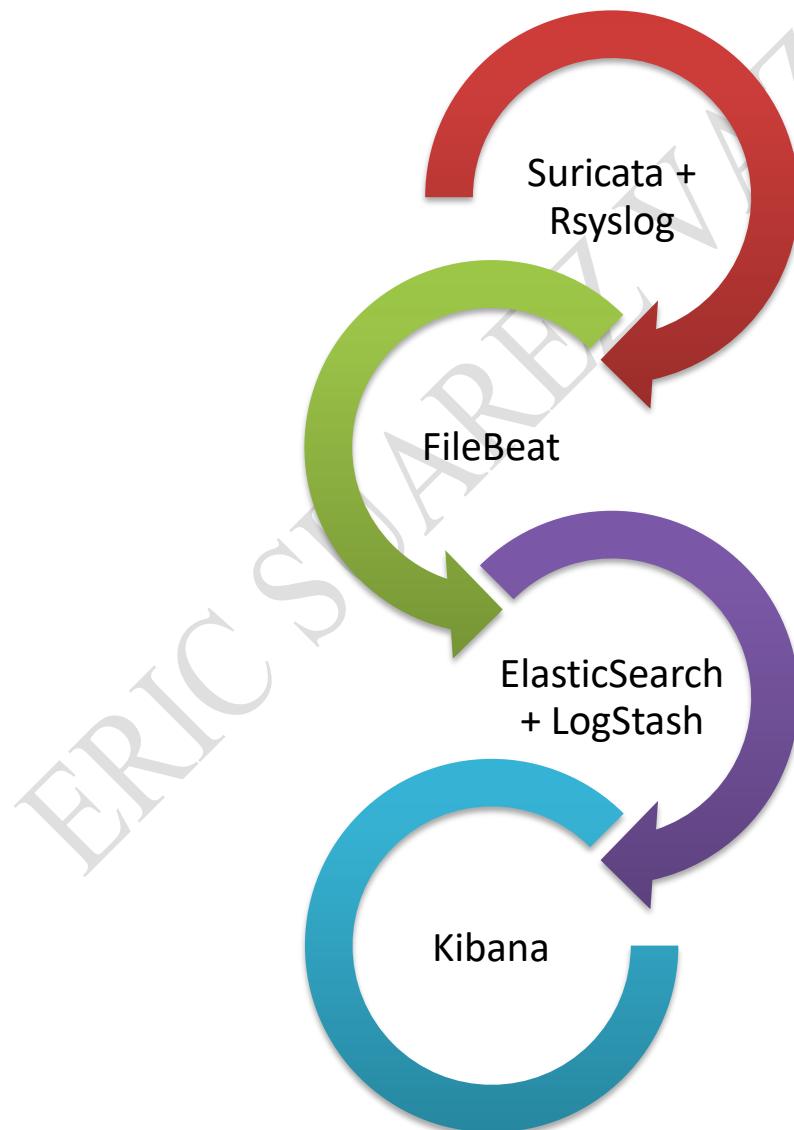
Objetivo principal del trabajo

El objetivo principal de este trabajo de **sistema gestor de eventos con ELK** es visualizar los registros **logs** que nuestra máquina genere ante un ataque o escaneo de manera sencilla visualmente hablando, sin necesidad de ir a los propios **logs** a mano.

¿Cómo lograremos esto?

El primer paso para elaborar este **SIEM** será instalar y configurar las herramientas de creación de registros **logs**, como por ejemplo **Suricata** y **Rsyslog** para que ante un ataque de fuerza bruta por **SSH** o un escaneo ya sea con **Nmap** o con la ejecución de un **Ping** se quede registrado en un **log**.

Estos logs serán mandados a hacia una base de datos creada por **ElasticSearch** y **Logstash** a manos de la herramienta **FileBeat** la cual recogerá dichos **logs** de las herramientas anteriores. Por último, usaremos **Kibana** para visualizar los **logs** recogidos por **FileBeat** y añadidos en la base de datos de **ElasticSearch**, donde podemos filtrar por los datos de dichos **logs**.



Instalaciones de Herramientas Necesarias

Con la máquina virtual ya instalada con la imagen anteriormente seleccionada, empezaremos a instalar todas las herramientas necesarias para el sistema gestor de eventos.

Aún no las configuraremos, solo las dejaremos instaladas.

Suricata

Con nuestra máquina nueva lista, ejecutaremos un **apt update** para actualizar todos los paquetes y acto seguido ejecutamos un **apt install suricata**

```
root@eric:/home/eric# apt update
Obj:1 http://es.archive.ubuntu.com/ubuntu mantic InRelease
Des:2 http://es.archive.ubuntu.com/ubuntu mantic-updates InRelease [109 kB]
Obj:3 http://security.ubuntu.com/ubuntu mantic-security InRelease
Obj:4 http://es.archive.ubuntu.com/ubuntu mantic-backports InRelease
Des:5 http://es.archive.ubuntu.com/ubuntu mantic-updates/restricted amd64 Packages [310 kB]
Descargados 418 kB en 1s (377 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se pueden actualizar 81 paquetes. Ejecute «apt list --upgradable» para verlos.
root@eric:/home/eric# apt install suricata
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libevent-2.1.7 libevent-core-2.1.7 libevent-pthreads-2.1.7 libhiredis0.14 libhttp2 libhyperscan5
  libluajit-5.1-2 libluajit-5.1-common libnet1 libnetfilter-log1 libnetfilter-queue1 oinkmaster
  snort-rules-default suricata-update
Paquetes sugeridos:
  snort | snort-pgsql | snort-mysql libtcmalloc-minimal4
Se instalarán los siguientes paquetes NUEVOS:
  libevent-2.1.7 libevent-core-2.1.7 libevent-pthreads-2.1.7 libhiredis0.14 libhttp2 libhyperscan5
  libluajit-5.1-2 libluajit-5.1-common libnet1 libnetfilter-log1 libnetfilter-queue1 oinkmaster
```

Esto instalará los paquetes necesarios de suricata, así como sus reglas por defecto, pero como aún no está validado, no creará las reglas de su librería (**/var/lib/suricata/rules**)

```
root@eric:/etc/suricata/rules# ls
app-layer-events.rules  files.rules      kerberos-events.rules  quic-events.rules  stream-events.rules
decoder-events.rules   ftp-events.rules  modbus-events.rules   rfb-events.rules   tls-events.rules
dhcp-events.rules     http2-events.rules mqtt-events.rules    smb-events.rules
dnp3-events.rules      http-events.rules nfc-events.rules     smtp-events.rules
dns-events.rules       ipsec-events.rules ntp-events.rules    ssh-events.rules
root@eric:/etc/suricata/rules# cd /var/lib/suricata/rules/
root@eric:/var/lib/suricata/rules# ls
root@eric:/var/lib/suricata/rules#
```

Lo que sí estará instalado es el archivo de configuración **suricata.yaml**, archivo que tocaremos más adelante cuando configuremos **Suricata**.

```
GNU nano 7.2
%YAML 1.1
---
# Suricata configuration file. In addition to the comments describing all
# options in this file, full documentation can be found at:
# https://docs.suricata.io/en/latest/configuration/suricata-yaml.html

# This configuration file generated by Suricata 7.0.0.
suricata-version: "7.0"

## Step 1: Inform Suricata about your network
##

##[ars:
# more specific is better for alert accuracy and performance
address-groups:
  HOME_NET: "[192.168.0.0/16,10.0.0.0/8,172.16.0.0/12]"
  #HOME_NET: "[192.168.0.0/16]"
  #HOME_NET: "[10.0.0.0/8]"
```

[2167 líneas leídas]

^G Ayuda **^O Guardar** **^W Buscar** **^K Cortar** **^T Ejecutar** **^C Ubicación** **M-U Deshacer**
^X Salir **^R Leer fich.** **^V Reemplazar** **^U Pegar** **^J Justificar** **^I Ir a línea** **M-E Rehacer**

Rsyslog

Para instalar esta herramienta tras instalar **suricata** volvemos a ejecutar un **apt update** para luego ejecutar un **apt install rsyslog**

```
root@eric:/home/eric# apt update
Obj:1 http://security.ubuntu.com/ubuntu mantic-security InRelease
Obj:2 http://es.archive.ubuntu.com/ubuntu mantic InRelease
Obj:3 http://es.archive.ubuntu.com/ubuntu mantic-updates InRelease
Obj:4 http://es.archive.ubuntu.com/ubuntu mantic-backports InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se pueden actualizar 81 paquetes. Ejecute «apt list --upgradable» para verlos.
root@eric:/home/eric# apt install rsyslog
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
rsyslog ya está en su versión más reciente (8.2306.0-2ubuntu2).
```

En mi caso, **rsyslog** ya estaba instalado en esta máquina por defecto, así que lo dejamos así y posteriormente entraremos en su configuración.

Filebeat

Como **FileBeat** pertenece a ElasticSearch, antes de instalarlo debemos ejecutar los siguientes comandos:

1. curl -fsSL <https://artifacts.elastic.co/GPG-KEY-elasticsearch> | sudo apt-key add -
2. echo "deb <https://artifacts.elastic.co/packages/7.x/apt> stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list

```
root@eric-ELK:/home/eric# curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
root@eric-ELK:/home/eric# echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list
deb https://artifacts.elastic.co/packages/7.x/apt stable main
root@eric-ELK:/home/eric#
```

Tras ejecutar estos comandos ejecutamos otro **update e install filebeat**

```
root@eric-ELK:/home/eric# apt update
Get:1 https://artifacts.elastic.co/packages/7.x/apt stable InRelease [13,6 kB]
Hit:2 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:4 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:5 https://artifacts.elastic.co/packages/7.x/apt stable/main i386 Packages [91,0 kB]
Get:6 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 Packages [129 kB]
Hit:7 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease
Fetched 352 kB in 1s (329 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
83 packages can be upgraded. Run 'apt list --upgradable' to see them.
W: https://artifacts.elastic.co/packages/7.x/apt/dists/stable/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
root@eric-ELK:/home/eric# apt install filebeat
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  filebeat
0 upgraded, 1 newly installed, 0 to remove and 83 not upgraded.
Need to get 36,9 MB of archives.
After this operation, 136 MB of additional disk space will be used.
Get:1 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 filebeat amd64 7.17.20 [36,9 MB]
Fetched 36,9 MB in 2s (23,3 MB/s)
Selecting previously unselected package filebeat.
(Reading database ... 200922 files and directories currently installed.)
Preparing to unpack .../filebeat_7.17.20_amd64.deb ...
Unpacking filebeat (7.17.20) ...
Setting up filebeat (7.17.20) ...
root@eric-ELK:/home/eric#
```

Con esto ya tendríamos FileBeat instalado y listo para configurar.

ElasticSearch

Para la instalación de **ElasticSearch**, debemos coger los paquetes necesarios para la instalación con otros comandos como hicimos con **FileBeat**, estos son los comandos que debemos ejecutar:

1. wget -qO - <https://artifacts.elastic.co/GPG-KEY-elasticsearch> | sudo apt-key add -
2. echo "deb <https://artifacts.elastic.co/packages/7.x/apt> stable main" | sudo tee /etc/apt/sources.list.d/elastic-7.x.list

```
root@eric-ELK:/home/eric# wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
gpg: can't open '-': No such file or directory
root@eric-ELK:/home/eric# echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee /etc/apt/sources.list.d/elastic-7.x.list
deb https://artifacts.elastic.co/packages/7.x/apt stable main
root@eric-ELK:/home/eric#
```

Estos paquetes también nos servirán para instalar **Logstash** y **Kibana** más adelante.

Tras obtener estos paquetes, ejecutamos otra vez un **apt update** e **install elasticsearch**

```
root@eric-ELK:/home/eric# apt update
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 https://artifacts.elastic.co/packages/7.x/apt stable InRelease
Hit:4 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:5 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
83 packages can be upgraded. Run 'apt list --upgradable' to see them.
W: https://artifacts.elastic.co/packages/7.x/apt/dists/stable/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
root@eric-ELK:/home/eric# apt install elasticsearch
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  elasticsearch
0 upgraded, 0 newly installed, 0 to remove and 83 not upgraded.
Need to get 327 MB of archives.
After this operation, 545 MB of additional disk space will be used.
Get:1 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 elasticsearch amd64 7.17.20 [327 MB]
Fetched 327 MB in 9s (36.0 MB/s)
Selecting previously unselected package elasticsearch.
(Reading database ... 202996 files and directories currently installed.)
Preparing to unpack .../elasticsearch_7.17.20_amd64.deb ...
Creating elasticsearch group... OK
Creating elasticsearch user... OK
Unpacking elasticsearch (7.17.20) ...
Setting up elasticsearch (7.17.20) ...
## NOT starting on installation, please execute the following statements to configure elasticsearch service to start automatically using systemd
sudo systemctl daemon-reload
sudo systemctl enable elasticsearch.service
## You can start elasticsearch service by executing
sudo systemctl start elasticsearch.service
Created elasticsearch keystore in /etc/elasticsearch/elasticsearch.keystore
root@eric-ELK:/home/eric#
```

Logstash

Aprovechamos los comandos anteriormente ejecutados e instalamos **logstash** con un **apt install logstash**

```
root@eric-ELK:/home/eric# apt install logstash
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  logstash
0 upgraded, 1 newly installed, 0 to remove and 83 not upgraded.
Need to get 367 MB of archives.
After this operation, 624 MB of additional disk space will be used.
Get:1 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 logstash amd64 1:7.17.20-1 [367 MB]
Fetched 367 MB in 11s (34.5 MB/s)
Selecting previously unselected package logstash.
(Reading database ... 204094 files and directories currently installed.)
Preparing to unpack .../logstash_1%3a7.17.20-1_amd64.deb ...
Unpacking logstash (1:7.17.20-1) ...
Setting up logstash (1:7.17.20-1) ...
Using bundled JDK: /usr/share/logstash/jdk
Using provided startup.options file: /etc/logstash/startup.options
OpenJDK 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be removed in a future release.
/usr/share/logstash/vendor/bundle/jruby/2.5.0/gems/pleaserun-0.0.32/lib/pleaserun/platform/base.rb:112: warning: constant ::Fixnum is deprecated
Successfully created system startup script for Logstash
root@eric-ELK:/home/eric#
```

Más adelante configuraremos esta herramienta combinándola con las demás

Kibana

Por último, e igual que con **Logstash**, aprovechamos los comandos anteriormente ejecutados e instalamos esta herramienta con un **apt install kibana**

```
root@eric-ELK:/home/eric# apt install kibana
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  kibana
0 upgraded, 1 newly installed, 0 to remove and 83 not upgraded.
Need to get 303 kB of archives.
After this operation, 781 kB of additional disk space will be used.
Get:1 https://artifacts.elastic.co/packages/7.x/apt/stable/main amd64 kibana amd64 7.17.20 [303 kB]
Fetched 303 kB in 8s (35,9 MB/s)
Selecting previously unselected package kibana.
(Reading database ... 219470 files and directories currently installed.)
Preparing to unpack .../kibana_7.17.20_amd64.deb ...
Unpacking kibana (7.17.20) ...
Setting up kibana (7.17.20) ...
Creating kibana group... OK
Creating kibana user... OK
Kibana is currently running with legacy OpenSSL providers enabled! For details and instructions on how to disable see https://www.elastic.co/guide/en/kibana/7.17/production.html#openssl-legacy-provider
Created Kibana keystore in /etc/kibana/kibana.keystore
root@eric-ELK:/home/eric#
```

Con esto ya tendríamos todas las herramientas instaladas y listas para su configuración.

Configuración de Herramientas

Configuración y reglas personales de Suricata

En el trabajo anterior de **Suricata** explicamos todo el proceso de configuración y uso de la propia herramienta, por ello, en este trabajo la explicación no será tan concreta, solo explicaré lo principal paso a paso de nuevo.

De primeras debemos saber nuestra interfaz de red con el siguiente comando

```
root@eric-ELK:/home/eric# ip -p -j route show default
[ {
    "dst": "default",
    "gatewav": "192.168.14.2",
    "dev": "ens33", highlighted
    "protocol": "dhcp",
    "metric": 100,
    "flags": [ ]
} ]
root@eric-ELK:/home/eric#
```

Nuestra interfaz de red es **ens33**

A continuación en **/etc/suricata/suricata.yaml** debemos cambiar **community-id** a **true**

```
GNU nano 6.2                                         /etc/suricata/suricata.yaml *
# enable/disable the community id feature.
community-id: true
# Seed value for the ID output. Valid values are 0-65535.
community-id-seed: 0
```

Y cambiar las interfaces que tengan puesto “**eth0**” a “**ens33**”

```
GNU nano 6.2                                         /etc/suricata/suricata.yaml *
format: "[%i] <%d> -- "
# type: json

##
## Step 3: Configure common capture settings
##
## See "Advanced Capture Options" below for more options, including Netmap
## and PF_RING.
##

# Linux high speed capture support
af-packet:
- interface: ens33 highlighted
# Number of receive threads. "auto" uses the number of cores
#threads: auto
# Default clusterid. AF_PACKET will load balance packets based on flow.
cluster-id: 99
```

¡Atención! Hay más de una en la configuración

Terminando con el **suricata.yaml** tenemos que indicar la ruta de las reglas de **suricata**

```
GNU nano 6.2                                         /etc/suricata/suricata.yaml
# When auto-config is enabled the hashmode specifies the algorithm for
# determining to which stream a given packet is to be delivered.
# This can be any valid Napatech NTPL hashmode command.
#
# The most common hashmode commands are: hash2tuple, hash2tuplesorted,
# hashtuple, hashStuplesorted and roundrobin.
#
# See Napatech NTPL documentation other hashmodes and details on their use.
#
# This parameter has no effect if auto-config is disabled.
#
hashmode: hashStuplesorted

##
## Configure Suricata to load Suricata-Update managed rules.
##

default-rule-path: /var/lib/suricata/rules
rule-files:
- suricata.rules highlighted

##
## Auxiliary configuration files.
##
```

Tras ello, actualizaremos **suricata** mediante los siguientes comandos

- **sudo suricata-update**
- **sudo suricata-update list-sources**
- **sudo suricata-update enable-source tgreen/hunting**

```
root@eric-ELK:/home/eric# suricata-update enable-source tgreen/hunting
10/4/2024 -- 16:56:23 - <Info> -- Using data-directory /var/lib/suricata.
10/4/2024 -- 16:56:23 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
10/4/2024 -- 16:56:23 - <Info> -- Using /etc/suricata/rules for Suricata provided rules.
10/4/2024 -- 16:56:23 - <Info> -- Found Suricata version 6.0.4 at /usr/bin/suricata.
10/4/2024 -- 16:56:23 - <Info> -- Creating directory /var/lib/suricata/update/sources
10/4/2024 -- 16:56:23 - <Info> -- Enabling default source et/open
10/4/2024 -- 16:56:23 - <Info> -- Source tgreen/hunting enabled
root@eric-ELK:/home/eric#
```

Por último, validaremos las configuraciones del **suricata** con este comando

- **sudo suricata -T -c /etc/suricata/suricata.yaml -v**

```
root@eric-ELK:/home/eric# suricata -T -c /etc/suricata/suricata.yaml -v
10/4/2024 -- 17:10:27 - <Info> - Running suricata under test mode
10/4/2024 -- 17:10:27 - <Notice> - This is Suricata version 6.0.4 RELEASE running in SYSTEM mode
10/4/2024 -- 17:11:37 - <Notice> - Configuration provided was successfully loaded. Exiting.
root@eric-ELK:/home/eric#
```

Con esto el **suricata** ya funciona con sus reglas predeterminadas, como queremos que **suricata** funcione en momentos específicos, queremos crear nuestras propias **reglas personales**, para ello creamos en **/var/lib/suricata/rules** nuestro propio **nano** de reglas.

```
GNU nano 6.2
alert icmp any any -> any any (msg:"ICMP Ping Detectado"; sid:100001;)
alert tcp any any -> any 22 (msg:"SSH Conexion Detectada"; sid:100002;)
```

Para que estos cambios funcionen, debemos añadir al **suricata.yaml** estas reglas en el **default-rule-path**

```
GNU nano 6.2
# This parameter has no effect if auto-config is disabled.
#
ports: [0-1,2-3]

# When auto-config is enabled the hashmode specifies the algorithm for
# determining to which stream a given packet is to be delivered.
# This can be any valid Napatech NTPL hashmode command.
#
# The most common hashmode commands are: hash2tuple, hash2tuplesorted,
# hash5tuple, hash5tuplesorted and roundrobin.
#
# See Napatech NTPL documentation other hashmodes and details on their use.
#
# This parameter has no effect if auto-config is disabled.
#
hashmode: hash5tuplesorted

##
## Configure Suricata to load Suricata-Update managed rules.
##

default-rule-path: /var/lib/suricata/rules

rule-files:
- suricata.rules
- personal.rules
```

Para guardar los cambios hechos, tenemos que volver a actualizar **suricata** con los comandos anteriores y volver a validar la configuración, por si acaso también reiniciaremos el **suricata** con **restart** y observamos que funcione con un **status**

```
root@eric-ELK:/home/eric# suricata -T -c /etc/suricata/suricata.yaml -v
10/4/2024 -- 17:28:57 - <Info> - Running suricata under test mode
10/4/2024 -- 17:28:57 - <Notice> - This is Suricata version 6.0.4 RELEASE running in SYSTEM mode
10/4/2024 -- 17:28:57 - <Info> - CPUs/cores online: 2
10/4/2024 -- 17:28:57 - <Info> - fast output device (regular) initialized: fast.log
10/4/2024 -- 17:28:57 - <Info> - eve-log output device (regular) initialized: eve.json
10/4/2024 -- 17:28:57 - <Info> - stats output device (regular) initialized: stats.log
10/4/2024 -- 17:27:05 - <Info> - 2 rule files processed. 37342 rules successfully loaded, 0 rules failed
10/4/2024 -- 17:27:05 - <Info> - Threshold config parsed: 0 rule(s) found
10/4/2024 -- 17:27:05 - <Info> - 37345 signatures processed. 1176 are IP-only rules, 5025 are inspecting packet payload, 30938 inspect application layer, 108 are decoder event only
10/4/2024 -- 17:28:03 - <Notice> - Configuration provided was successfully loaded. Exiting.
10/4/2024 -- 17:28:03 - <Info> - cleaning up signature grouping structure... complete
root@eric-ELK:/home/eric# Segunda validacion
```

Ya tenemos nuestras propias reglas de **suricata** establecidas, estas se guardan en un registro **log** llamado **fast.log**, a continuación ejecutaremos un **ping** desde otra máquina personal a la máquina donde estamos montando el **SIEM** para probar que el **suricata** funciona.

```
root@eric-ELK:/var/lib/suricata/rules# systemctl restart suricata
root@eric-ELK:/var/lib/suricata/rules# systemctl status suricata
● suricata.service - Suricata IDS/IDP daemon
   Loaded: loaded (/lib/systemd/system/suricata.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2024-04-10 18:31:15 CEST; 6s ago
     Docs: man:suricata(8)
           man:suricatasc(8)
           https://suricata-ids.org/docs/
   Process: 3949 ExecStart=/usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml --pidfile /run/suricata.pid (code=exited, status=0/SUCCESS)
   Main PID: 3951 (Suricata-Main)
      Tasks: 1 (limit: 4554)
     Memory: 251.3M
        CPU: 6.542s
       CGroup: /system.slice/suricata.service
               └─3951 /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml --pidfile /run/suricata.pid

abr 10 18:31:15 eric-ELK systemd[1]: Starting Suricata IDS/IDP daemon...
abr 10 18:31:15 eric-ELK suricata[3949]: 10/4/2024 -- 18:31:15 - <Notice> - This is Suricata version 6.0.4 RELEASE running in SYSTEM mode
abr 10 18:31:15 eric-ELK systemd[1]: Started Suricata IDS/IDP daemon.
root@eric-ELK:/var/lib/suricata/rules# ifconfig
ens33: flags=413<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.14.137 brd 192.168.14.255 netmask 255.255.255.0 broadcast 192.168.14.255
        inet6 fe80::ae10:7000:7a1c:7604 brd fe80::fffe:7000:7a1c:7604 scopeid 0x20<link>
            ether 00:0c:29:74:bb:3a txqueuelen 1000 (Ethernet)
            RX packets 5591 bytes 7365230 (7.3 MB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 2082 bytes 191639 (191.6 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 brd 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 brd ::1 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
            RX packets 242 bytes 23646 (23.6 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 242 bytes 23646 (23.6 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
root@eric-ELK:/var/lib/suricata/rules# cat personal.rules
alert icmp any any -> 192.168.14.0/24 any (msg:"paquetes ICMP de PING Capturados"; sid:1000000; rev:1;)
alert tcp any any -> 192.168.14.0/24 22 (msg:"conexion SSH Detectada"; flow:to_server; app-layer-protocol:ssh; sid:2271009; rev:1;)
root@eric-ELK:/var/lib/suricata/rules# cat /var/log/suricata/fast.log
04/10/2024-18:32:25.558317 [**] [1:1:000000:1] paquetes ICMP de PING Capturados [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.14.130:8 -> 192.168.14.137:0
04/10/2024-18:32:25.558349 [**] [1:1:000000:1] paquetes ICMP de PING Capturados [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.14.137:0 -> 192.168.14.130:0
root@eric-ELK:/var/lib/suricata/rules#
```

Ya hemos terminado con las primeras configuraciones del **suricata**, a medida que vayamos configurando otras herramientas debemos ir actualizando también las anteriores, como esta, pero la configuración principal está terminada.

Configuración de Rsyslog

Tras configurar la herramienta **suricata**, vamos a configurar nuestra **máquina virtual** como **máquina cliente de rsyslog** para que nos elabore los registros **logs** de dicha máquina virtual, para posteriormente añadir esos registros al **elasticsearch**.

En Ubuntu 22.04, la herramienta **rsyslog** ya viene preinstalada como hemos comprobado anteriormente, para configurarla debemos entrar en su **archivo.conf** e indicar la **ip del cliente (nuestra ip local)**

Así que entramos a la configuración con nano /etc/rsyslog.conf , descomentamos la configuración **UDP** y **TCP** y configuramos nuestra ip local a los posibles protocolos (**UDP** – con una @) o (**TCP** – con dos @), en este caso nuestro protocolo será **TCP** y este se comunicará por el **puerto 514**.

Además le indicaremos donde guardar los registros **logs**.

```
GNU nano 6.2
# /etc/rsyslog.conf configuration file for rsyslog
#
# For more information install rsyslog-doc and see
# /usr/share/doc/rsyslog-doc/html/configuration/index.html
#
# Default logging rules can be found in /etc/rsyslog.d/50-default.conf

#####
#### MODULES #####
#####

module(load="imuxsock") # provides support for local system logging
#module(load="immark") # provides --MARK-- message capability

# provides UDP syslog reception
module(load="imudp")
input(type="imudp" port="514")

# provides TCP syslog reception
module(load="imtcp")
input(type="imtcp" port="514")

$template RemoteLogs,"/var/log/remote/%HOSTNAME%/%PROGRAMNAME%.log"
.*?RemoteLogs
.*?@192.168.14.137:514
```

Tras elaborar esta configuración reiniciamos el servicio **rsyslog** con un **service restart rsyslog** y vemos su status

```
root@eric-ELK:/home/eric# nano /etc/rsyslog.conf
root@eric-ELK:/home/eric# service rsyslog restart
root@eric-ELK:/home/eric# service rsyslog status
● rsyslog.service - System Logging Service
   Loaded: loaded (/lib/systemd/system/rsyslog.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2024-04-10 19:30:59 CEST; 4s ago
     TriggeredBy: ● syslog.socket
     Docs: man:rsyslogd(8)
           man:rsyslog.conf(5)
           https://www.rsyslog.com/doc/
   Main PID: 3518 (rsyslogd)
     Tasks: 10 (limit: 4554)
    Memory: 110.8M
       CPU: 3.980s
      CGroup: /system.slice/rsyslog.service
              └─3518 /usr/sbin/rsyslogd -n -iNONE

abr 10 19:30:59 eric-ELK systemd[1]: Starting System Logging Service...
abr 10 19:30:59 eric-ELK rsyslogd[3518]: imuxsock: Acquired UNIX socket '/run/systemd/journal/syslog' (fd 3) from systemd. [v8.2112.0]
abr 10 19:30:59 eric-ELK systemd[1]: Started System Logging Service.
abr 10 19:30:59 eric-ELK rsyslogd[3518]: rsyslogd's groupid changed to 111
abr 10 19:30:59 eric-ELK rsyslogd[3518]: rsyslogd's userid changed to 104
abr 10 19:30:59 eric-ELK rsyslogd[3518]: [origin software="rsyslogd" swVersion="8.2112.0" x-pid="3518" x-info="https://www.rsyslog.com"] start
root@eric-ELK:/home/eric#
```

Ejecutamos un logger para comprobar que rsyslog funciona y guarda los registros log.

Antes de continuar, hay que saber que **rsyslog** no usa un **solo archivo.log** sino que usa varios en función del **hostname** y el **programname**, por ello habrá varios **archivos.logs**, (**¡Y se irán creando más con el uso de diferentes aplicaciones!**) todos ellos los intentaremos añadir en el **elasticsearch**, pero los importantes son el **root.log** (registro de movimientos del usuario root), el **rsyslog.log** (registros log de la propia herramienta rsyslog) y el **systemd.log** (registro de logs del sistema)

Configuración de ElasticSearch

Abrimos el archivo de configuración de **ElasticSearch** con **sudo nano**

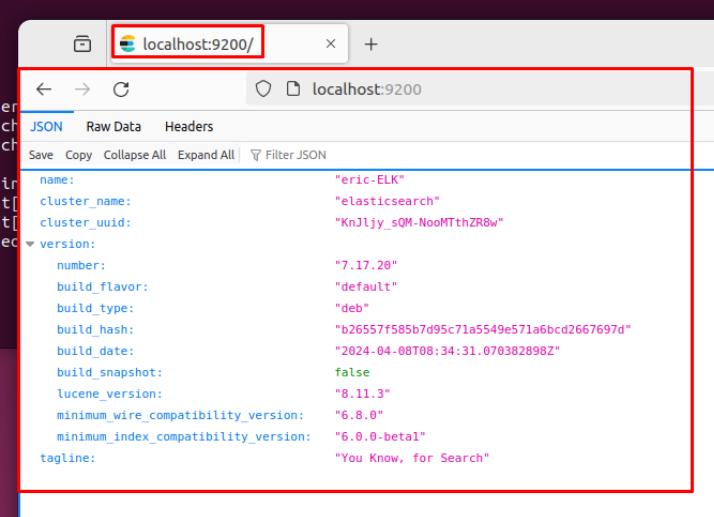
/etc/elasticsearch/elasticsearch.yml

En el descomentamos el **network.host**, lo cambiamos a localhost y el puerto por el que se escucha **ElasticSearch** lo dejamos en el 9200

```
GNU nano 6.2                               /etc/elasticsearch/elasticsearch.yml
# ----- Network -----
#
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
#
network.host: localhost
#
# By default Elasticsearch listens for HTTP traffic on the first free port it
# finds starting at 9200. Set a specific HTTP port here:
#
http.port: 9200
#
```

Ejecutamos un **systemctl start elasticsearch** y a los pocos minutos se ejecutará el servicio, para comprobar que este funciona, usaremos **Firefox** para irnos a nuestro **localhost** por el puerto **9200** y ver el resultado

```
root@eric-ELK:/home/eric# nano /etc/elasticsearch/elasticsearch.yml
root@eric-ELK:/home/eric# systemctl start elasticsearch
root@eric-ELK:/home/eric# systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
   Loaded: loaded (/lib/systemd/system/elasticsearch.service; disabled; vendor preset: enabled)
   Active: active (running) since Thu 2024-04-11 04:46:47 CEST; 8min ago
     Docs: https://www.elastic.co
 Main PID: 4915 (java)
    Tasks: 65 (limit: 4554)
   Memory: 1.6G
      CPU: 1min 4.924s
     CGroup: /system.slice/elasticsearch.service
             └─4915 /usr/share/elasticsearch
                  ├─5097 /usr/share/elasticsearch
root@eric-ELK:/home/eric# [redacted]
[redacted]
```



name:	"eric-ELK"																				
cluster_name:	"elasticsearch"																				
cluster_uuid:	"KnJljy_sQM-NooMTthZRBw"																				
version:	<table border="1"><thead><tr><td>number:</td><td>"7.17.20"</td></tr></thead><tbody><tr><td>build_flavor:</td><td>"default"</td></tr><tr><td>build_type:</td><td>"deb"</td></tr><tr><td>build_hash:</td><td>"b26557f585b7d95c71a5549e571a6bcd2667697d"</td></tr><tr><td>build_date:</td><td>"2024-04-08T08:34:31.070382898Z"</td></tr><tr><td>build_snapshot:</td><td>false</td></tr><tr><td>lucene_version:</td><td>"8.11.3"</td></tr><tr><td>minimum_wire_compatibility_version:</td><td>"6.8.0"</td></tr><tr><td>minimum_index_compatibility_version:</td><td>"6.0.0-beta1"</td></tr><tr><td>tagline:</td><td>"You Know, for Search"</td></tr></tbody></table>	number:	"7.17.20"	build_flavor:	"default"	build_type:	"deb"	build_hash:	"b26557f585b7d95c71a5549e571a6bcd2667697d"	build_date:	"2024-04-08T08:34:31.070382898Z"	build_snapshot:	false	lucene_version:	"8.11.3"	minimum_wire_compatibility_version:	"6.8.0"	minimum_index_compatibility_version:	"6.0.0-beta1"	tagline:	"You Know, for Search"
number:	"7.17.20"																				
build_flavor:	"default"																				
build_type:	"deb"																				
build_hash:	"b26557f585b7d95c71a5549e571a6bcd2667697d"																				
build_date:	"2024-04-08T08:34:31.070382898Z"																				
build_snapshot:	false																				
lucene_version:	"8.11.3"																				
minimum_wire_compatibility_version:	"6.8.0"																				
minimum_index_compatibility_version:	"6.0.0-beta1"																				
tagline:	"You Know, for Search"																				

Realmente, la configuración base de **elasticsearch** “acaba aquí” ya que ahora empezaríamos por la parte gráfica (el **dashboard**) con **Kibana** aunque este archivo lo tocaremos de nuevo en un futuro.

Configuración de Kibana

Antes de empezar con la configuración de **Kibana** ejecutaremos un **systemctl start** y **status** para comprobar que inicia perfectamente y entraremos en el **localhost** por el puerto de **Kibana** que es el **5601**, esta página nos indicará que **Kibana** funciona pero no está aún configurado.

```
root@eric-ELK:/home/eric# systemctl start kibana
root@eric-ELK:/home/eric# systemctl status kibana
● kibana.service - Kibana
   Loaded: loaded (/etc/systemd/system/kibana.service; disabled; vendor preset: enabled)
   Active: active (running) since Thu 2024-04-11 22:42:03 CEST; 8s ago
     Docs: https://www.elastic.co
 Main PID: 4689 (node)
    Tasks: 7 (limit: 4554)
   Memory: 101.6M
      CPU: 4.378s
     CGroup: /system.slice/kibana.service
             └─4689 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/bin/../src/cli/dist --loggin>
```

abr 11 22:42:03 eric-ELK systemd[1]: Starting Kibana...
abr 11 22:42:03 eric-ELK kibana[4689]: [main] 2024-04-11T22:42:03.378Z | info | [main] Starting Kibana v8.10.0 (build 36DALo41FR1QMO)
abr 11 22:42:03 eric-ELK kibana[4689]: [main] 2024-04-11T22:42:03.378Z | info | [main] Node.js v18.15.0
abr 11 22:42:03 eric-ELK kibana[4689]: [main] 2024-04-11T22:42:03.378Z | info | [main] OS Linux 5.15.0-102-generic x64
abr 11 22:42:03 eric-ELK kibana[4689]: [main] 2024-04-11T22:42:03.378Z | info | [main] Memory 101.6M
abr 11 22:42:03 eric-ELK kibana[4689]: [main] 2024-04-11T22:42:03.378Z | info | [main] CPU 4.378s
abr 11 22:42:03 eric-ELK kibana[4689]: [main] 2024-04-11T22:42:03.378Z | info | [main] CGroup /system.slice/kibana.service
abr 11 22:42:03 eric-ELK kibana[4689]: [main] 2024-04-11T22:42:03.378Z | info | [main] └─4689 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/bin/../src/cli/dist --loggin>

localhost:5601

Kibana server is not ready yet

Empezaremos la configuración de **Kibana** creando un usuario llamado “eric” en mi caso y guardándolo en el archivo **htpasswd.users** de la carpeta **Nginx** con el siguiente comando

`echo "eric:‘openssl passwd -apr1’" | sudo tee -a /etc/nginx/htpasswd.users`

```
root@eric-ELK:/home/eric# echo "eric:‘openssl passwd -apr1’" | sudo tee -a /etc/nginx/htpasswd.users
Password:
Verifying - Password:
eric:$apr1$I7qQ0S/g$5dasRRW36DALo41FR1QMO
root@eric-ELK:/home/eric#
```

Esto pedirá una contraseña (que en mi caso será igual que el usuario, eric)

Ya tenemos un usuario de **Kibana** instalado, ahora, crearemos un servidor con **Nginx** (ya instalado de antes en la máquina virtual, si no está instalado se instala con **apt install nginx-core**)

Crearemos un **nano** en **/etc/nginx/sites-available/elk** (**elk** es una carpeta que vamos a crear ahora, el nombre es arbitrario)

En dicho archivo, escribiremos lo siguiente:

```
GNU nano 6.2                               /etc/nginx/sites-available/elk
server {
    listen 80;

    server_name elk;

    auth_basic "Restricted Access";
    auth_basic_user_file /etc/nginx/htpasswd.users;

    location / {
        proxy_pass http://localhost:5601;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

Cuando ya tengamos el archivo creado activaremos la configuración creando un link simbólico en la carpeta “**sites-available**” con el siguiente comando:

```
sudo ln -s /etc/nginx/sites-available/elk /etc/nginx/sites-enabled/elk
```

```
root@eric-ELK:/etc/nginx/sites-available# ln -s /etc/nginx/sites-available/elk /etc/nginx/sites-enabled/elk
root@eric-ELK:/etc/nginx/sites-available#
```

Tras la configuración, ejecutamos un **nginx -t** para corroborar que la sintaxis es adecuada y no hay errores, y reiniciamos el servicio **nginx**

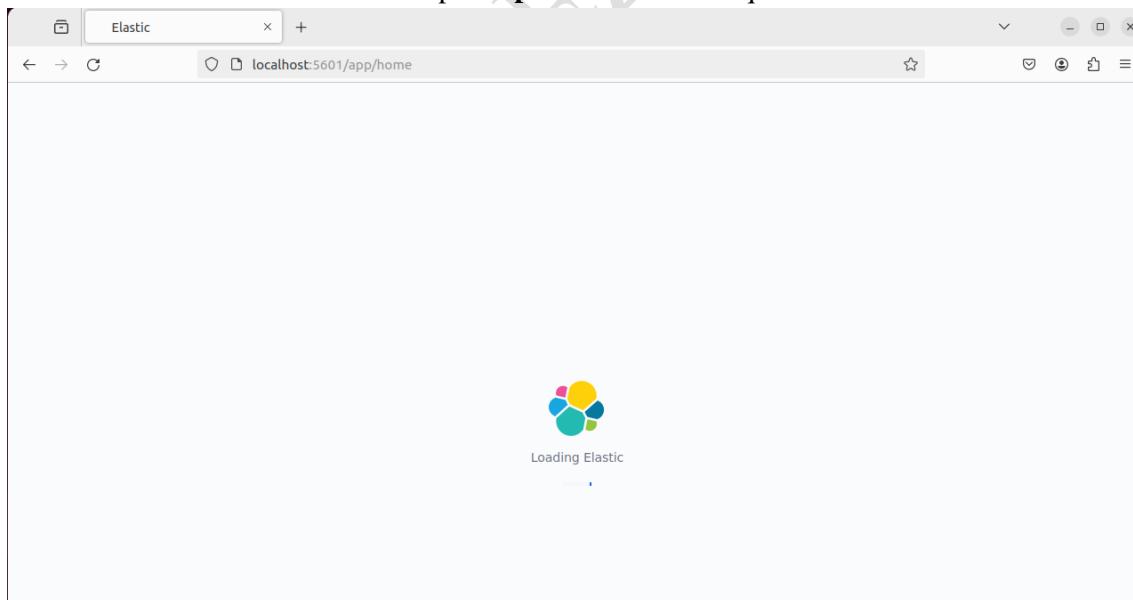
```
root@eric-ELK:/etc/nginx/sites-available# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
root@eric-ELK:/etc/nginx/sites-available# systemctl reload nginx
root@eric-ELK:/etc/nginx/sites-available#
```

Por último, debemos permitir el paso de **Nginx** por nuestro Firewall mediante el comando **sudo ufw allow 'Nginx Full'**

```
root@eric-ELK:/etc/nginx/sites-available# ufw allow 'Nginx Full'
Rules updated
Rules updated (v6)
root@eric-ELK:/etc/nginx/sites-available#
```

Página de ElasticSearch

Con esto, gran parte de la configuración de **Kibana** ya está terminada, vamos a comprobar su éxito entrando de nuevo al localhost por el **puerto de Kibana** que es el **5601**



Nos empezará a cargar **ElasticSearch** y nos saltará un inicio de integraciones

The screenshot shows a web browser window with the URL `localhost:5601/app/home#/` highlighted by a red box. The page title is "Welcome to Elastic". Below the title is a colorful illustration of data visualization components like charts and graphs. A call-to-action box titled "Start by adding integrations" contains the text: "Add data to your cluster from any source, then analyze and visualize it in real time. Use our solutions to add search anywhere, observe your ecosystem, and protect against security threats." It features two buttons: "Add integrations" (highlighted with a red box) and "Explore on my own". At the bottom, there's a note about usage data collection and links to the Privacy Statement and a link to disable it.

Podemos instalar módulos ahora o después, el caso es que ya tendremos nuestro servidor de **ElasticSearch** operativo

The screenshot shows a web browser window with the URL `localhost:5601/app/home#/` highlighted by a red box. The page title is "Welcome home". It features four main modules: "Enterprise Search" (yellow card), "Observability" (pink card), "Security" (teal card), and "Analytics" (blue card). The "Analytics" module is highlighted with a red box. Below the modules is a section titled "Get started by adding integrations" with instructions to start working with data using ingest options like collecting data from an app or service, or uploading a file. It includes three buttons: "Add integrations" (highlighted with a red box), "Try sample data", and "Upload a file". To the right of the "Analytics" card is another colorful illustration of data visualization components.

SIEM con ELK – Eric Suárez Vázquez – Incidentes de ciberseguridad

The screenshot shows the Elasticsearch Stack Management interface. On the left, a sidebar navigation includes sections for Management, Ingest, Data, Alerts and Insights, and Kibana. Under Kibana, the 'Index Patterns' section is selected. The main content area is titled 'Index patterns' and contains a search bar and a table header 'Pattern ↑'. A banner at the top of the main content area reads: 'Create and manage the index patterns that help you retrieve your data from Elasticsearch.' Below the table header, it says 'No items found'.

The 'Kibana status' section is highlighted with a red box. It displays various performance metrics:

- 1.94 GB** (Heap total)
- 196.71 MB** (Heap used)
- 1.40** (Requests per second)
- 1.20, 1.03, 1.06** (Load)
- 10.92 ms** (Delay avg)
- 39.67 ms** (Response time avg)
- 1m; 5m; 15m** (Load interval)
- 50: 10.49 ms; 95: 11.02 ms; 99: 25.30 ms** (Percentiles)
- 111.00 ms** (Response time max)

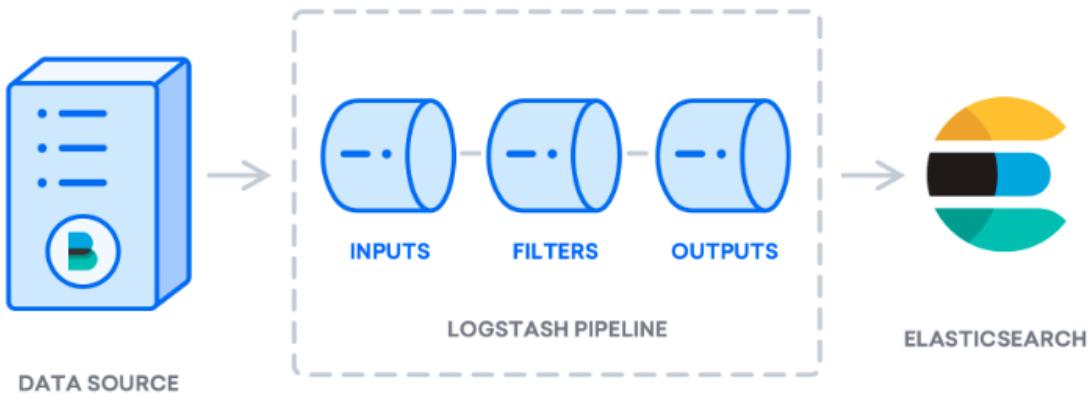
The 'Plugin status' section shows the status of various Elasticsearch plugins:

ID	Status
core:elasticsearch	Elasticsearch is available
core:savedObjects	SavedObjects service has completed migrations and is available
plugin:licensing	License fetched
plugin:banners	All dependencies are available
plugin:features	All dependencies are available
plugin:globalSearch	All dependencies are available
plugin:globalSearchProviders	All dependencies are available
plugin:uiActionsEnhanced	All dependencies are available
plugin:embeddableEnhanced	All dependencies are available
plugin:urlDrilldown	All dependencies are available

A banner at the bottom right of the Kibana status section reads: '⚠ Your data is not secure. Don't lose one bit. Enable our free security features.' with options to 'Enable security' or 'Dismiss'.

Configuración de Logstash

La página principal de **ElasticSearch** junto con **Kibana** funciona, ahora pasaremos a configurar **Logstash**. Esta herramienta sirve para recoger los datos obtenidos por **Filebeat** y transformarlo a un formato más común para después enviarlo a la base de datos de **ElasticSearch**.



Empecemos la configuración de esta herramienta, primero de todo entramos en la carpeta raíz de la configuración de **Logstash** que es `/etc/logstash/conf.d/` es esta carpeta crearemos un archivo mediante nano que indicará al **FileBeat**s que puerto usar (el **Filebeat**s lo configuraremos más adelante)

```

root@eric-ELK: /etc/logstash/conf.d#
root@eric-ELK: /etc/logstash/conf.d# nano 02-beats-input.conf
GNU nano 6.2
input {
  beats {
    port => 5044
  }
}

```

Ya tenemos porqué logstash cogerá los archivos logs de **Filebeat**, a continuación crearemos **otro archivo de configuración** que se ocupará de mandar estos archivos logs a la base de datos del **ElasticSearch**

```

root@eric-ELK: /etc/logstash/conf.d#
root@eric-ELK: /etc/logstash/conf.d# nano 30-elasticsearch-output.conf
GNU nano 6.2
output {
  if [@metadata][pipeline] {
    elasticsearch {
      hosts => ["localhost:9200"]
      manage_template => false
      index => "%{@metadata}[beat]-%{@metadata}[version]-%{+YYYY.MM.dd}"
      pipeline => "%{@metadata}[pipeline]"
    }
  } else {
    elasticsearch {
      hosts => ["localhost:9200"]
      manage_template => false
      index => "%{@metadata}[beat]-%{@metadata}[version]-%{+YYYY.MM.dd}"
    }
  }
}

```

Esta configuración muestra el puerto que usa **ElasticSearch (9200)** y el mensaje del **index** que nos mostrará la página web con la fecha en formato **años.meses.días**

Comprobamos que la configuración que hemos añadido no tiene errores con el siguiente comando: `sudo -u logstash /usr/share/logstash/bin/logstash—path.settings /etc/logstash -t`

```

root@eric-ELK:/etc/logstash/conf.d# -u logstash /usr/share/logstash/bin/logstash --path.settings /etc/logstash -t
-u: command not found
root@eric-ELK:/etc/logstash/conf.d# sudo -u logstash /usr/share/logstash/bin/logstash -path.settings /etc/logstash -t
Using bundled JDK: /usr/share/logstash/jdk
OpenJDK 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be removed in a future release.
Sending Logstash logs to /var/log/logstash which is now configured via log4j2.properties
[2024-04-12T02:41:22,149][INFO ][logstash.runner] Log4j configuration path used is: /etc/logstash/log4j2.properties
[2024-04-12T02:41:22,270][INFO ][logstash.runner] Starting Logstash ("logstash.version">"7.17.20", "jruby.version">="jruby 9.2.20.1 (2.5
.8) 2021-11-30 2a2962fb1 OpenJDK 64-Bit Server VM 11.0.22+7 on 11.0.22+7 +Indy +jit [linux-x86_64]")
[2024-04-12T02:41:22,274][INFO ][logstash.runner] JVM bootstrap flags: [-Xms1g, -Xmx1g, -XX:+UseConcMarkSweepGC, -XX:CMSInitiatingOccupan
cyFraction=75, -XX:+UseCMSInitiatingOccupancyOnly, -Djava.awt.headless=true, -Dfile.encoding=UTF-8, -Djdk.io.File.enableADS=true, -Djruby.compile.i
nvokedynamic=true, -Djruby.jit.threshold=0, -Djruby.regexp.interruptible=true, -XX:+HeapDumpOnOutOfMemoryError, -Djava.security.egd=file:/dev/urand
om, -Dlog4j2.isThreadContextMapInheritable=true]
[2024-04-12T02:41:22,334][INFO ][logstash.settings] Creating directory [:setting=>"path.queue", :path=>"/var/lib/logstash/queue"]
[2024-04-12T02:41:22,356][INFO ][logstash.settings] Creating directory [:setting=>"path.dead_letter_queue", :path=>"/var/lib/logstash/dead_
letter_queue"]
[2024-04-12T02:41:26,153][INFO ][org.reflections.Reflections] Reflections took 218 ms to scan 1 urls, producing 119 keys and 419 values
Configuration OK
[2024-04-12T02:41:27,832][INFO ][logstash.runner] Using config.test_and_exit mode. Config Validation Result: OK. Exiting Logstash
root@eric-ELK:/etc/logstash/conf.d#

```

Nota: Aunque el OpenJDK esté deprecado, sigue siendo usable

Por último ejecutamos un **sudo systemctl start logstash** y un **status** para comprobar que funciona el servicio.

```

root@eric-ELK:/etc/logstash/conf.d# systemctl start logstash
root@eric-ELK:/etc/logstash/conf.d# systemctl status logstash
● logstash.service - logstash
   Loaded: loaded (/etc/systemd/system/logstash.service; disabled; vendor preset: enabled)
   Active: active (running) since Fri 2024-04-12 02:45:38 CEST; 3s ago
     Main PID: 4764 (java)
       Tasks: 14 (limit: 4554)
      Memory: 162.6M
        CPU: 5.745s
       CGroup: /system.slice/logstash.service
               └─4764 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=75 -XX:+UseCMSInitia
abr 12 02:45:38 eric-ELK systemd[1]: Started logstash.
abr 12 02:45:38 eric-ELK logstash[4764]: Using bundled JDK: /usr/share/logstash/jdk
abr 12 02:45:38 eric-ELK logstash[4764]: OpenJDK 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely b
Lines 1-13/13 (END)

```

Configuración de FileBeat

Pasamos a la última configuración de herramientas configurando el **FileBeat**, su archivo de configuración se encuentra en **/etc/filebeat/filebeat.yml**

En nuestro caso, **Filebeat** no va a mandar datos logs a **ElasticSearch**, sino a **Logstash** y este, si lo mandará formateado a **ElasticSearch**, así que debemos comentar las líneas que conectan **FileBeat** con **ElasticSearch** y descomentar las líneas que conectan **FileBeat** con **Logstash**

```

GNU nano 6.2
/etc/filebeat/filebeat.yml *
# ===== Outputs =====
# Configure what output to use when sending the data collected by the beat.

# ----- Elasticsearch Output -----
#output.elasticsearch:
#  # Array of hosts to connect to.
#  # hosts: ["localhost:9200"]

#  # Protocol - either `http` (default) or `https`.
#  #protocol: "https"

#  # Authentication credentials - either API key or username/password.
#  #api_key: "id:api_key"
#  #username: "elastic"
#  #password: "changeme"

# ----- Logstash Output -----
#output.logstash:
#  # The Logstash hosts
#  hosts: ["localhost:5044"]

# Optional SSL. By default is off.
# List of root certificates for HTTPS server verifications
#ssl.certificateAuthorities: ["/etc/pki/root/ca.pem"]

# Certificate for SSL client authentication
#ssl.certificate: "/etc/pki/client/cert.pem"

# Client Certificate Key
#ssl.key: "/etc/pki/client/cert.key"

```

Tras cambiar esta configuración activaremos los módulos de **FileBeat** con el sistema, y posteriormente listaremos los módulos disponibles con los siguientes comandos

```
root@eric-ELK:/home/eric# filebeat modules enable system
Enabled system
root@eric-ELK:/home/eric# filebeat modules enable system
Module system is already enabled
root@eric-ELK:/home/eric# filebeat modules list
Enabled:
system

Disabled:
activemq
apache
auditd
aws
awsfargate
azure
barracuda
bluecoat
cef
checkpoint
cisco
coredns
crowdstrike
cyberark
cyberarkpas
cylance
elasticsearch
envoyproxy
f5
fortinet
gcp
google_workspace
googlecloud
gsuite
haproxy
ibmmq
icinga
iis
imperva
infoblox
iptables
juniper
kafka
kibana
logstash
microsoft
misp
mongodb
```

Esto nos sacará un listado de los módulos activados y los desactivados, de la misma manera, activaremos también el suricata con **FileBeat**

```
root@eric-ELK:/home/eric# filebeat modules enable suricata
Enabled suricata
root@eric-ELK:/home/eric#
```

No se nos debe olvidar añadir los registros **Logs de Suricata** al **FileBeat** indicando de donde debe cogerlos, esto lo ponemos en la configuración en **/etc/filebeat/filebeat.yml**

```
GNU nano 6.2                                         /etc/filebeat/filebeat.yml

# For more available modules and options, please see the filebeat.reference.yml sample
# configuration file.

# ===== Filebeat inputs =====

filebeat.inputs:

# Each - is an input. Most options can be set at the input level, so
# you can use different inputs for various configurations.
# Below are the input specific configurations.

# filestream is an input for collecting log messages from files.
- type: filestream

  # Unique ID among all inputs, an ID is required.
  id: my-filestream-id

  # Change to true to enable this input configuration.
  enabled: false

  # Paths that should be crawled and fetched. Glob based paths.
  paths:
    - /var/log/*.log
    #- c:\programdata\elasticsearch\logs\*

#Agregamos los logs de Suricata
- type: log
  enabled: true
  paths:
    - /var/log/suricata/fast.log
```

Continuamos ejecutando el siguiente código que nos ayudará a **parsear** los datos de los archivos **logs** antes de mandárselo a **Logstash** y este a **ElasticSearch**

```
root@eric-ELK:/home/eric# nano /etc/filebeat/filebeat.yml
root@eric-ELK:/home/eric# sudo filebeat setup --pipelines --modules system suricata
```

A continuación cargaremos un template o una plantilla de **Filebeat** con el siguiente comando

```
root@eric-ELK:/home/eric# sudo filebeat setup --index-management -E output.logstash.enabled=false -E 'output.elasticsearch.hosts=["localhost:9200"]'
Overwriting ILM policy is disabled. Set setup.tlm.overwrite: true for enabling.

Index setup finished.
root@eric-ELK:/home/eric#
```

Hacemos lo mismo pero añadiendo **Kibana** con el siguiente comando

```
root@eric-ELK:/home/eric# sudo filebeat setup -E output.logstash.enabled=false -E output.elasticsearch.hosts=['localhost:9200'] -E setup.kibana.host=localhost:5601
Overwriting ILM policy is disabled. Set setup.tlm.overwrite: true for enabling.

Index setup finished.
Loading dashboards (Kibana must be running and reachable)
Loaded dashboards
Setting up ML using setup --machine-learning is going to be removed in 8.0.0. Please use the ML app instead.
See more: https://www.elastic.co/guide/en/machine-learning/current/index.html
It is not possible to load ML jobs into an Elasticsearch 8.0.0 or newer using the Beat.
Loaded machine learning job configurations
Loaded Ingest pipelines
root@eric-ELK:/home/eric#
```

Atencion: **Filebeat** deja de ser usable con **Elasticsearch 8.0.0** y superior, por ello es normal que nos de error y puede que no funcione, igualmente explicaremos los pasos a seguir para que funcione si tuviéramos una versión anterior, en este caso, sí nos funciona.

Reiniciamos el servicio de **FileBeat** y mediante comando ejecutamos una consulta con **FileBeat**, si nos devuelve datos significa que ha obtenido datos de **ElasticSearch**

```
root@eric-ELK:/home/eric# systemctl start filebeat
root@eric-ELK:/home/eric# curl -XGET 'http://localhost:9200/filebeat-*/_search?pretty'
{
  "took" : 34,
  "timed_out" : false,
  "_shards" : {
    "total" : 2,
    "successful" : 2,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 10000,
      "relation" : "gte"
    },
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "filebeat-7.17.20-2024.04.12",
        "_type" : "_doc",
        "_id" : "swEi044BfVkr24NxXnm",
        "_score" : 1.0,
        "_ignored" : [
          "event.original"
        ],
        "_source" : {
          "agent" : {
            "hostname" : "eric-ELK",
            "name" : "eric-ELK",
            "id" : "f8e69e5c-1a10-4124-996d-0e91aac30d0d",
            "ephemeral_id" : "cf67f4e3-ea76-42cb-9835-308986d95cad",
            "type" : "filebeat",
            "version" : "7.17.20"
          }
        },
        "log" : {
```

Página ElasticSearch

Entreamos a nuestro servidor de **ElasticSearch** y clickamos en **Kibana** para ver los datos que **FileBeat** y **Logstash** han traído a nuestro servidor

The screenshot shows the Kibana home page with a dark header bar. The URL in the address bar is `localhost:5601/app/home#/`. Below the header, there's a search bar with the placeholder "Search Elastic". The main content area has a title "Welcome home" and four cards:

- Enterprise Search**: Create search experiences with a refined set of APIs and tools.
- Observability**: Consolidate your logs, metrics, application traces, and system availability with purpose-built UIs.
- Security**: Prevent, collect, detect, and respond to threats for unified protection across your infrastructure.
- Analytics**: Explore, visualize, and analyze your data using a powerful suite of analytical tools and applications.

The "Analytics" card is highlighted with a red border.

Kibana tiene muchas utilidades con respecto a la visualización de los datos, los más interesantes son el “**Dashboard**” y “**Discover**”

The screenshot shows the Kibana Analytics page with a dark header bar. The URL in the address bar is `localhost:5601/app/kibana_overview#/`. Below the header, there's a search bar with the placeholder "Search Elastic". The main content area has a title "Analytics" and two large cards:

- Dashboard**: Analyze data in dashboards.
- Discover**: Search and find insights.

Both the "Dashboard" and "Discover" cards are highlighted with red borders. Below these are three smaller cards:

- Canvas**: Design pixel-perfect presentations.
- Maps**: Plot geographic data.
- Machine Learning**: Model, predict, and detect.

At the top right of the Analytics page, there are links for "Dev tools", "Manage", and "Add integrations".

SIEM con ELK – Eric Suárez Vázquez – Incidentes de ciberseguridad

Nos centraremos principalmente en el apartado **Discover**, en él observaremos todos los archivos logs ordenados por más recientes sacados por la herramienta **FileBeat** y parseado por **Logstash**

The screenshot shows the Elastic Discover interface with the URL `localhost:5601/app/discover?/_g=(filters:(),refreshInterval:(pause:lt,value:0),time:(from:'2024-04-12T16:33:30.000Z',to:'2024-04-12T16:34:00))`. The search bar contains the query `filebeat-*`. The results show 245 hits. A histogram at the top indicates event distribution over time. The main table lists log entries, with the first few lines shown below:

```
> Apr 12, 2024 @ 18:33:57.616 @timestamp: Apr 12, 2024 @ 18:33:57.616 @version: 1 agent.ephemeral_id: cf67f4e3-ea76-42cb-9835-308986d95cad agent.hostname: eric-ELK agent.id: f8e69e5c-1a10-4124-996d-0e91aac30d0d agent.name: eric-ELK agent.type: filebeat agent.version: 7.17.20 ecs.version: 1.12.0 event.category: network event.created: Apr 12, 2024 @ 18:33:58.328 event.dataset: suricata.eve.event.ingested: Apr 12, 2024 @ 18:33:59.433 event.kind: metric event.module: suricata event.original: {"@timestamp": "2024-04-12T18:33:57.616274+0200", "event_type": "stats", "stats": {"uptime": 8734, "capture": {"kernel_packets": 8768, "kernel_drops": 0, "errors": 0}, "decoder": {"pkts": 8768, "bytes": 6278015, "invalid": 0, "ipv4": 8080, "ipv6": 501, "ether": 8768, "chdlc": 0, "raw": 0, "null": 0, "tcp": 7094, "udp": 1462, "sctp": 0, "icmpv4": 0, "icmpv6": 0}}, "flow_id": "177963620953911", "in_iface": "ens33", "event_type": "fileinfo", "src_ip": "2.21.39.17", "src_port": 80, "time": "2024-04-12T18:33:56.038000+0200"}, "flow_id": "177963620953911", "in_iface": "ens33", "event_type": "fileinfo", "src_ip": "2.21.39.17", "src_port": 80, "time": "2024-04-12T18:33:54.224000+0200"}, "flow_id": "177963620953911", "in_iface": "ens33", "event_type": "fileinfo", "src_ip": "2.21.39.17", "src_port": 80, "time": "2024-04-12T18:33:54.214000+0200"}>
```

En adición, tenemos una barra buscadora donde podemos filtrar los registros **logs**, podemos filtrarlos por el nombre “**suricata**” para comprobar que efectivamente obtiene los **logs** de **suricata**

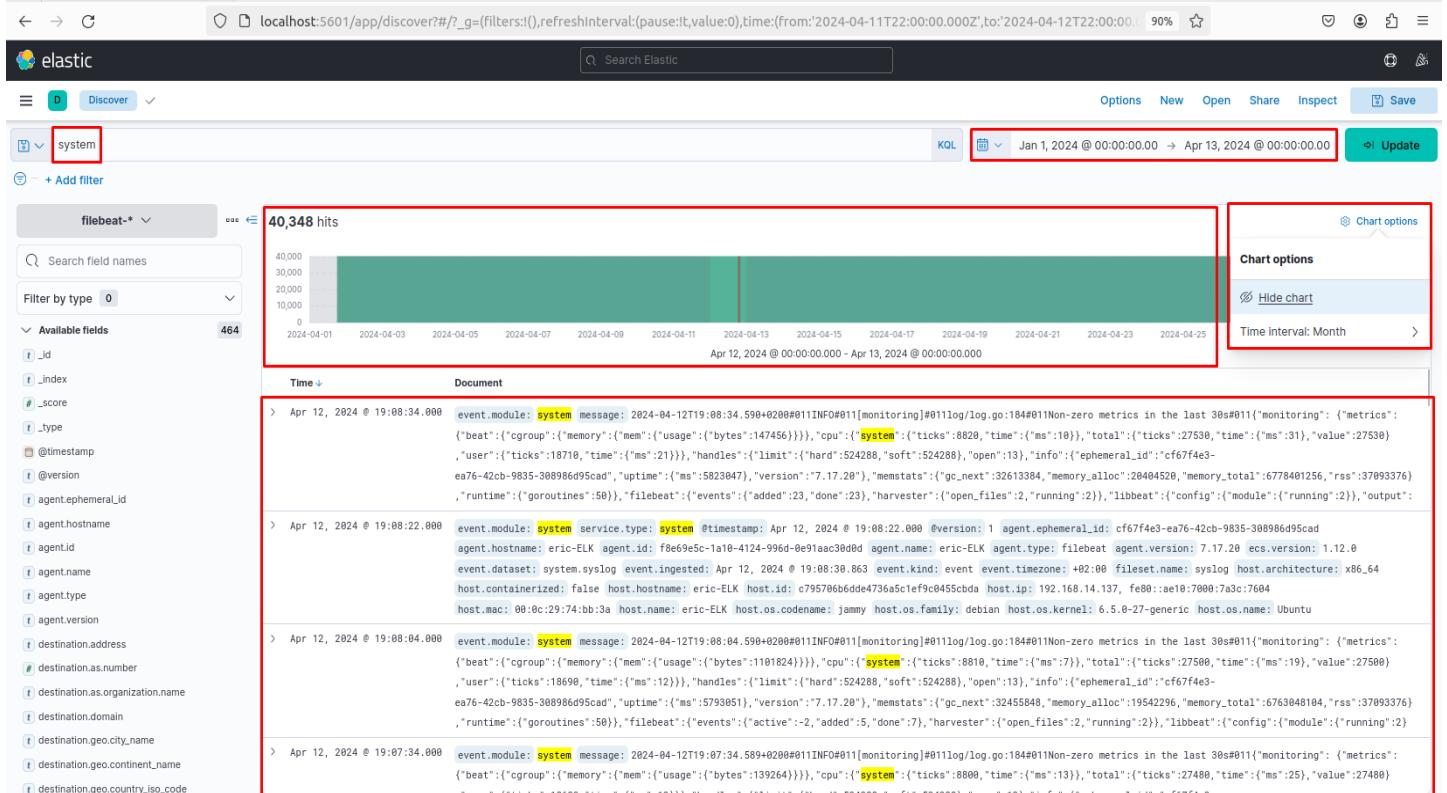
The screenshot shows the Elastic Discover interface with the URL `localhost:5601/app/discover?/_g=(filters:(),refreshInterval:(pause:lt,value:0),time:(from:'2024-04-12T16:33:30.000Z',to:'2024-04-12T16:34:00))`. The search bar contains the query `suricata`. The results show 214 hits. A histogram at the top indicates event distribution over time. The main table lists log entries, with the first few lines shown below:

```
> Apr 12, 2024 @ 18:33:57.616 event.module: suricata service.type: suricata tags: suricata, beats_input_raw_event @timestamp: Apr 12, 2024 @ 18:33:57.616 @version: 1 agent.ephemeral_id: cf67f4e3-ea76-42cb-9835-308986d95cad agent.hostname: eric-ELK agent.id: f8e69e5c-1a10-4124-996d-0e91aac30d0d agent.name: eric-ELK agent.type: filebeat agent.version: 7.17.20 ecs.version: 1.12.0 event.category: network event.created: Apr 12, 2024 @ 18:33:58.328 event.dataset: suricata.eve.event.ingested: Apr 12, 2024 @ 18:33:59.433 event.kind: metric event.original: {"@timestamp": "2024-04-12T18:33:57.616274+0200", "event_type": "stats", "stats": {"uptime": 8734, "capture": {"kernel_packets": 8768, "kernel_drops": 0, "errors": 0}, "decoder": {"pkts": 8768, "bytes": 6278015, "invalid": 0, "ipv4": 8080, "ipv6": 501, "ether": 8768, "chdlc": 0, "raw": 0, "null": 0, "tcp": 7094, "udp": 1462, "sctp": 0, "icmpv4": 0, "icmpv6": 0}}, "flow_id": "177963620953911", "in_iface": "ens33", "event_type": "fileinfo", "src_ip": "2.21.39.17", "src_port": 80, "time": "2024-04-12T18:33:56.038000+0200"}, "flow_id": "177963620953911", "in_iface": "ens33", "event_type": "fileinfo", "src_ip": "2.21.39.17", "src_port": 80, "time": "2024-04-12T18:33:54.224000+0200"}, "flow_id": "177963620953911", "in_iface": "ens33", "event_type": "fileinfo", "src_ip": "2.21.39.17", "src_port": 80, "time": "2024-04-12T18:33:54.214000+0200"}>
```

SIEM con ELK – Eric Suárez Vázquez – Incidentes de ciberseguridad

Al igual que hemos filtrado por suricata, este poderoso conjunto de herramientas puede filtrar los archivos logs de toda aplicación instalada y todo lo ocurrido en el sistema, al igual que puede filtrar por distintas fechas seleccionadas

En este ejemplo, hemos filtrado por todos los archivos **logs** del sistema filtrando por el nombre “**system**” y filtrando también por el día 1 de Enero de este año, es decir, filtramos todos los logs desde el inicio del año hasta ahora.



Este comando saca más de 40.000 registros diferentes en una línea de tiempo que es modificable mediante el botón “**Chart options**”, pudiendo modificar dicha línea por intervalos de años, meses, días u horas.

SIEM con ELK – Eric Suárez Vázquez – Incidentes de ciberseguridad

Kibana tiene también una función de mostrar los últimos registros logs de hace 15 minutos ordenados por los más nuevos.

The screenshot shows the Kibana Discover interface. The search bar contains 'filebeat-*'. The results count is 1,529 hits. A red box highlights the 'Time' dropdown set to 'Last 15 minutes'. The main area features a histogram with teal bars representing the number of hits per minute from 01:20:00 to 01:34:00. Below the histogram is a table of log documents, each with a timestamp, version, agent information, destination details, event duration, and original log entry. The table is scrollable and shows several entries, with the last one being: 'Apr 13, 2024 @ 01:34:23.393 @timestamp: Apr 13, 2024 @ 01:34:23.393 @version: 1 agent.ephemeral_id: 10240b6a-cd66-47b3-bf45-f22313714dc agent.hostname: eric-ELK agent.id: f8e69e5c-1a10-4124-996d-0e91aac30dd0 agent.name: eric-ELK agent.type: filebeat agent.version: 7.17.20 destination.address: 192.168.178.2 destination.bytes: 2798 destination.ip: 192.168.178.2 destination.packets: 1 destination.port: 53 ecs.version: 1.12.0 event.category: network event.created: Apr 13, 2024 @ 01:34:24.454 event.dataset: suricata.eve event.duration: 12.0 event.end: Apr 13, 2024 @ 01:28:27.312 event.ingested: Apr 13, 2024 @ 01:34:25.609 event.kind: event event.module: suricata event.original: {"timestamp": "2024-04-13T01:34:24.134691+0200", "flow_id": "1093471890543647",

Una vez terminado con el apartado **Discover** explicaremos el apartado **Dashboard**.

Kibana tiene por defecto varios dashboards creados, aunque también podemos crear el nuestro personalizado.

The screenshot shows the Kibana Dashboard interface. A red box highlights the 'Create dashboard' button in the top right corner. The main area lists various pre-created dashboards with their titles, descriptions, tags, and actions. The dashboards include: [Filebeat AWS] CloudTrail, [Filebeat AWS] ELB Access Log Overview, [Filebeat AWS] S3 Server Access Log Overview, [Filebeat AWS] VPC Flow Log Overview, [Filebeat ActiveMQ] Application Events, [Filebeat ActiveMQ] Audit Events, [Filebeat Apache] Access and error logs ECS, [Filebeat Auditd] Audit Events ECS, [Filebeat Azure] Alerts Overview, [Filebeat Azure] Cloud Overview, and [Filebeat Azure] User Activity.

SIEM con ELK – Eric Suárez Vázquez – Incidentes de ciberseguridad

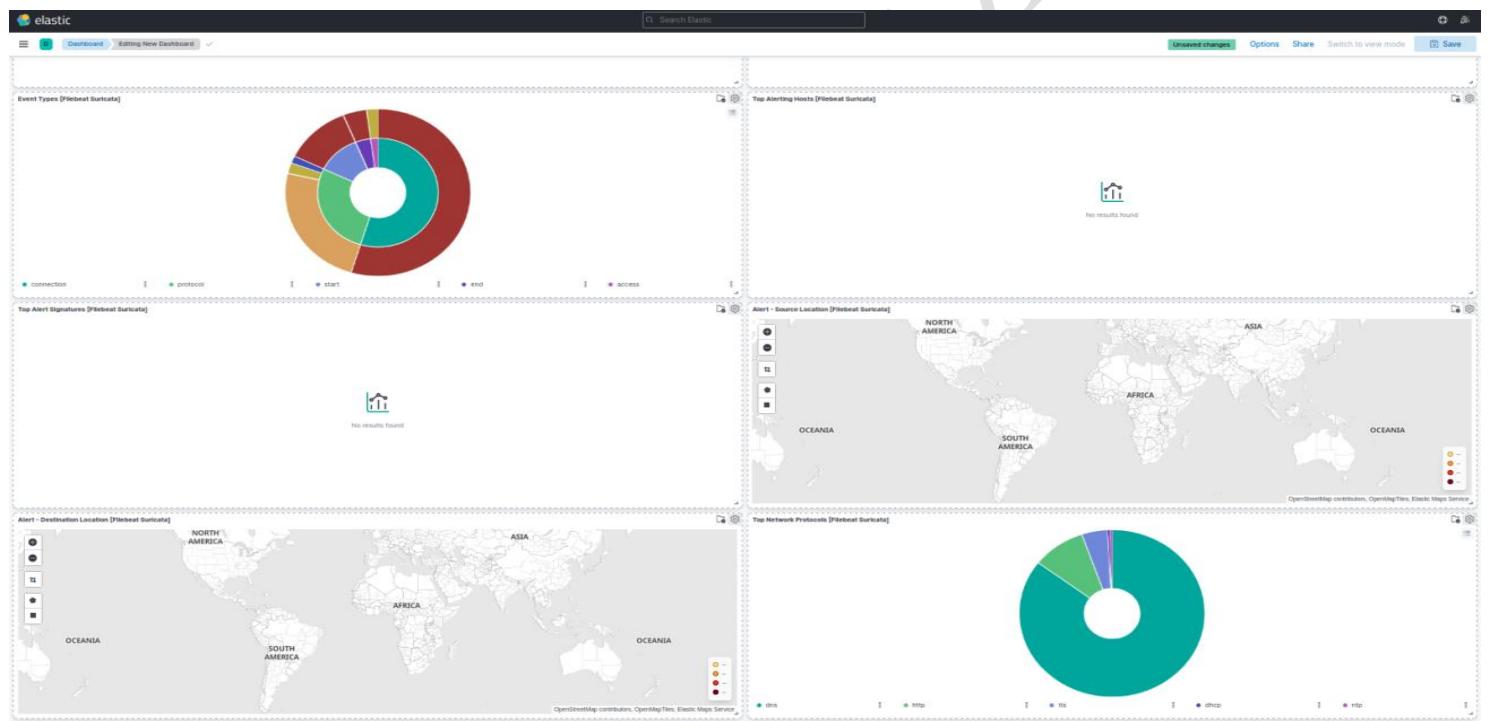
Dentro de nuestro dashboard añadimos filtros de **suricata** para ver los datos de estos logs

The screenshot shows the Kibana interface with a search bar at the top containing 'suricata'. A red box highlights the 'Add from library' button in the top right corner of the main dashboard area. To the right, a modal window titled 'Add from library' is open, also containing a search bar with 'suricata'. Below it is a list of ten items, each preceded by a small icon and the text '[Filebeat Suricata]':

- Navigation [Filebeat Suricata]
- Event Types [Filebeat Suricata]
- Event Count [Filebeat Suricata]
- Top Alerting Hosts [Filebeat Suricata]
- Top Alert Signatures [Filebeat Suricata]
- Alert - Source Location [Filebeat Suricata]
- Alert - Destination Location [Filebeat Suricata]
- Top Network Protocols [Filebeat Suricata]
- Top Transport Protocols [Filebeat Suricata]
- Alerts [Filebeat Suricata]

At the bottom of the modal, there is a 'Rows per page: 10' dropdown.

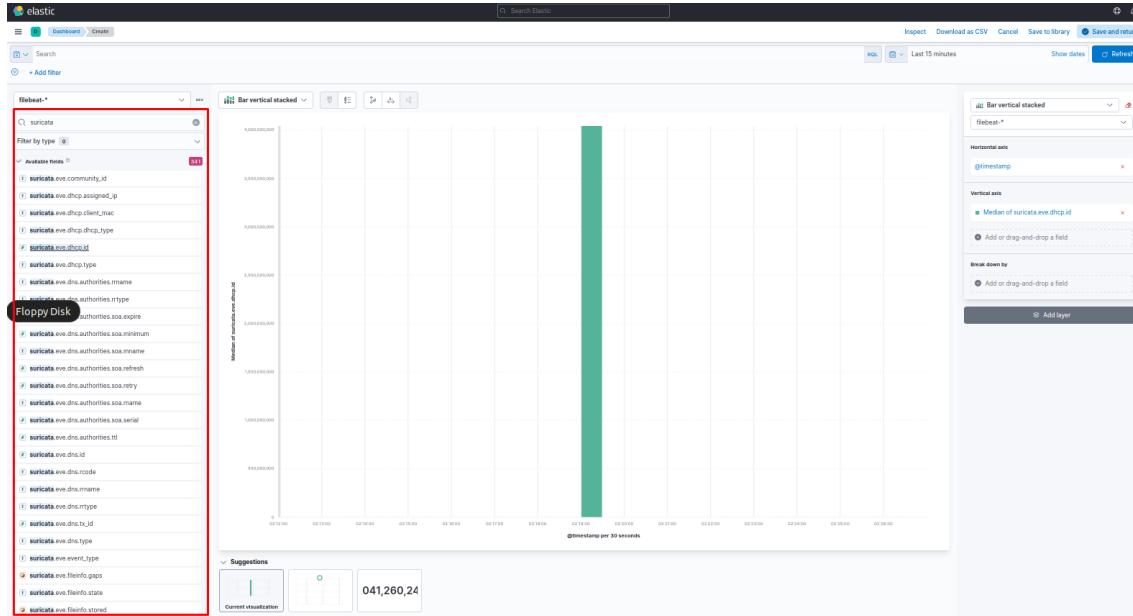
Estos son datos obtenidos por **suricata** que podemos filtrar y observarlo en forma gráfica



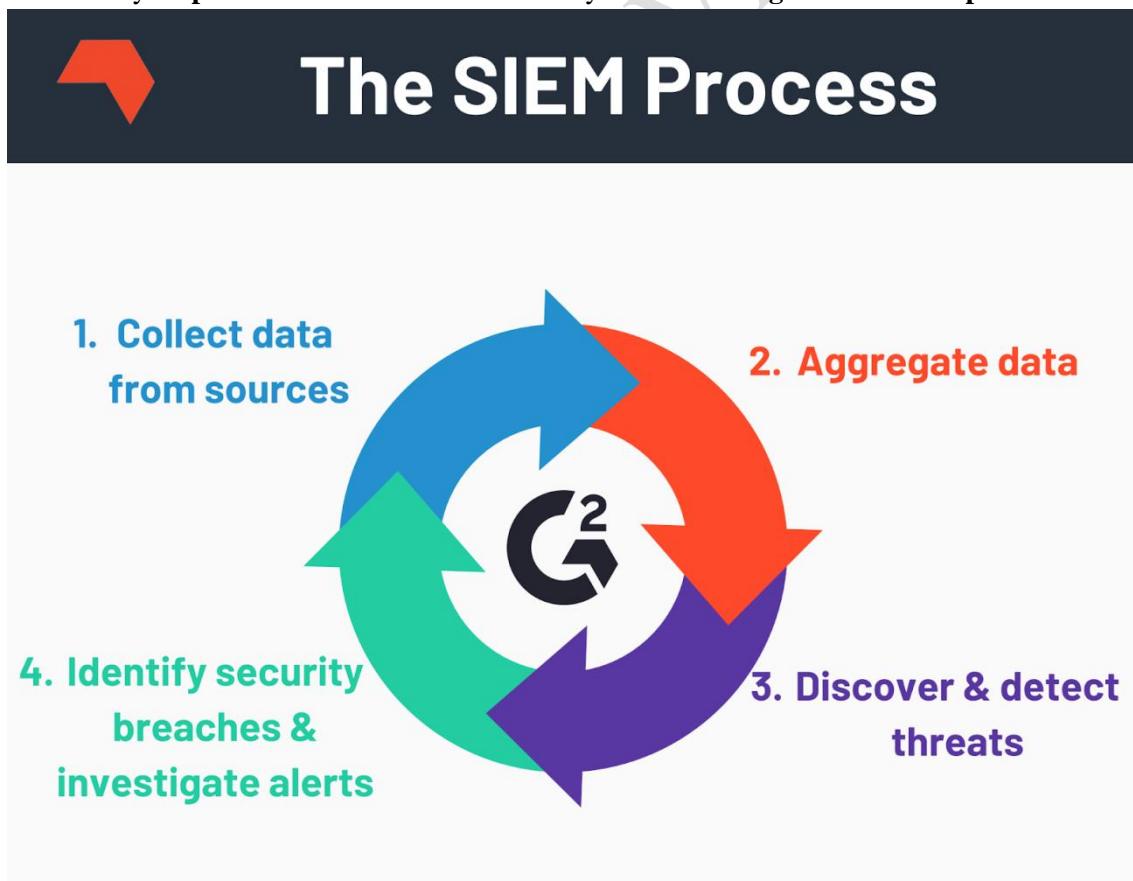
Con esta herramienta podríamos crear miles de diferentes combinaciones, y no solo con **suricata** como **IDS** sino con diferentes herramientas o los propios logs del sistema

Nota: Este gestor de evento consume muchos recursos, yo he experimentado una gran lentitud a la hora de manipular **ElasticSearch** y **Kibana**

Además podemos elaborar nuestra propia gráfica eligiendo los logs que queramos



Con todo este trabajo elaborado, hemos aprendido que un **SIEM** con **ELK** es una herramienta esencial para mejorar la postura de seguridad de una organización, proporcionando **visibilidad, detección y respuesta efectivas** a las amenazas y eventos de seguridad en tiempo real.



Securización del ELK

Plugin

Para empezar con esto, debemos instalar un par de plugins para **elasticsearch** y **kibana**

NOTA: También se pueden descargar en .zip y descomprimir el plugin:

<https://www.elastic.co/guide/en/elasticsearch/reference/6.2/installing-xpack-es.html>

Install X-Pack

1 Install X-Pack into Elasticsearch

```
bin/elasticsearch-plugin install x-pack
```

<https://www.elastic.co/guide/en/kibana/6.2/installing-xpack-kb.html>

4 Install X-Pack into Kibana

```
bin/kibana-plugin install x-pack
```

Credenciales de inicio

Con nuestro servidor **ELK** ya montado y funcional, vamos a intentar **securizarlo**, de primeras vamos añadir un **usuario y contraseña** a nuestro servidor para que solo nosotros o las personas a las que demos permiso podamos entrar.

Para ello configuraremos el archivo de configuración de **elasticsearch**, en él añadiremos la siguiente línea y reiniciaremos **elasticsearch**

```
[GNU nano 6.2] [ /etc/elasticsearch/elasticsearch.yml ]  
# ----- Various -----  
  
# Require explicit names when deleting indices:  
#  
#action.destructive_requires_name: true  
#  
# ----- Security -----  
#  
# *** WARNING ***  
#  
# Elasticsearch security features are not enabled by default.  
# These features are free, but require configuration changes to enable them.  
# This means that users don't have to provide credentials and can get full access  
# to the cluster. Network connections are also not encrypted.  
#  
# To protect your data, we strongly encourage you to enable the Elasticsearch security features.  
# Refer to the following documentation for instructions.  
#  
# https://www.elastic.co/guide/en/elasticsearch/reference/7.16/configuring-stack-security.html  
xpack.security.enabled: true  
[ Wrote 97 lines ]  
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo  
^X Exit ^R Read File ^A Replace ^U Paste ^J Justify ^I Go To Line M-E Redo  
M-A Set Mark M-6 Copy
```

El siguiente paso será “pedirle” contraseñas a **elasticsearch** con el comando, esto generará contraseñas que debemos guardar para todas las herramientas

```
root@eric-ELK:/home/eric# ./usr/share/elasticsearch/bin/elasticsearch-setup-passwords auto  
Initiating the setup of passwords for reserved users elastic,apm_system,kibana,kibana_system,logstash_system,beats_system,remote_monitoring_user.  
The passwords will be randomly generated and printed to the console.  
Please confirm that you would like to continue [y/N]y  
  
Changed password for user apm_system  
PASSWORD apm_system = 3W0n8eB5iZ52akzr3XgN  
  
Changed password for user kibana_system  
PASSWORD kibana_system = Ooqi6gD7UiJaKhU4DS43  
  
Changed password for user kibana  
PASSWORD kibana = Ooqi6gD7UiJaKhU4DS43  
  
Changed password for user logstash_system  
PASSWORD logstash_system = IPLnhdgjV2lD9TndKuoT  
  
Changed password for user beats_system  
PASSWORD beats_system = fIPilgad2RcRS3Kp0b0o  
  
Changed password for user remote_monitoring_user  
PASSWORD remote_monitoring_user = HIpp0lb4VJ26ZGwvqMg4  
  
Changed password for user elastic  
PASSWORD elastic = T1QgkNHWp009IW52RLbH  
  
root@eric-ELK:/home/eric#
```

A continuación entramos en la configuración de **Kibana**, descomentamos las líneas y cambiamos la contraseña por la autogenerada por el anterior comando, luego reiniciamos **Kibana**

```
GNU nano 6.2
# The Kibana server's name. This is used for display purposes.
#server.name: "your-hostname"

# The URLs of the Elasticsearch instances to use for all your queries.
#elasticsearch.hosts: ["http://localhost:9200"]

# Kibana uses an index in Elasticsearch to store saved searches, visualizations and
# dashboards. Kibana creates a new index if the index doesn't already exist.
#kibana.index: ".kibana"

# The default application to load.
#kibana.defaultAppId: "home"

# If your Elasticsearch is protected with basic authentication, these settings provide
# the username and password that the Kibana server uses to perform maintenance on the Kibana
# index at startup. Your Kibana users still need to authenticate with Elasticsearch, which
# is proxied through the Kibana server.
elasticsearch.username: "elk"
elasticsearch.password: "T10gkNHwp009IW52RLbH"
```

Al intentar entrar en **elasticsearch** por el puerto **9200**, nos pedirá autenticación

 **localhost:9200**

This site is asking you to sign in.

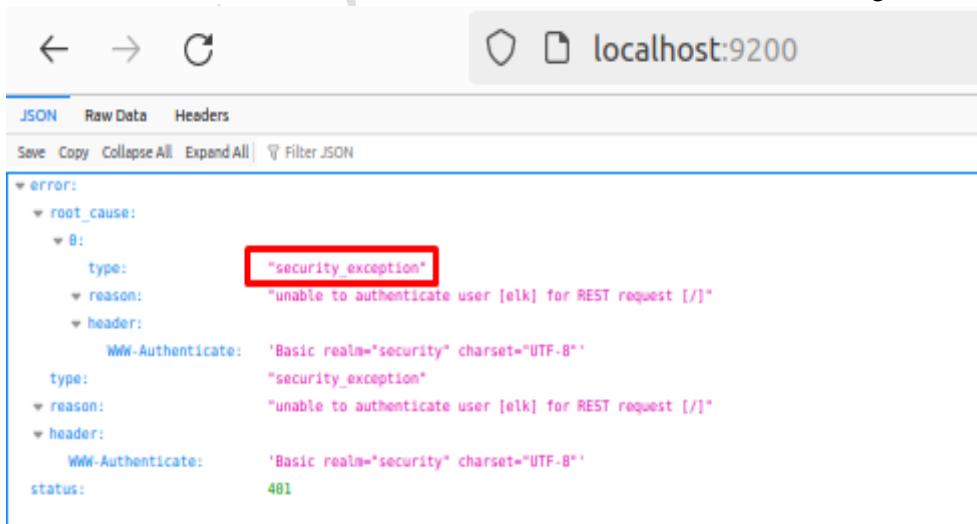


Username
elk

Password

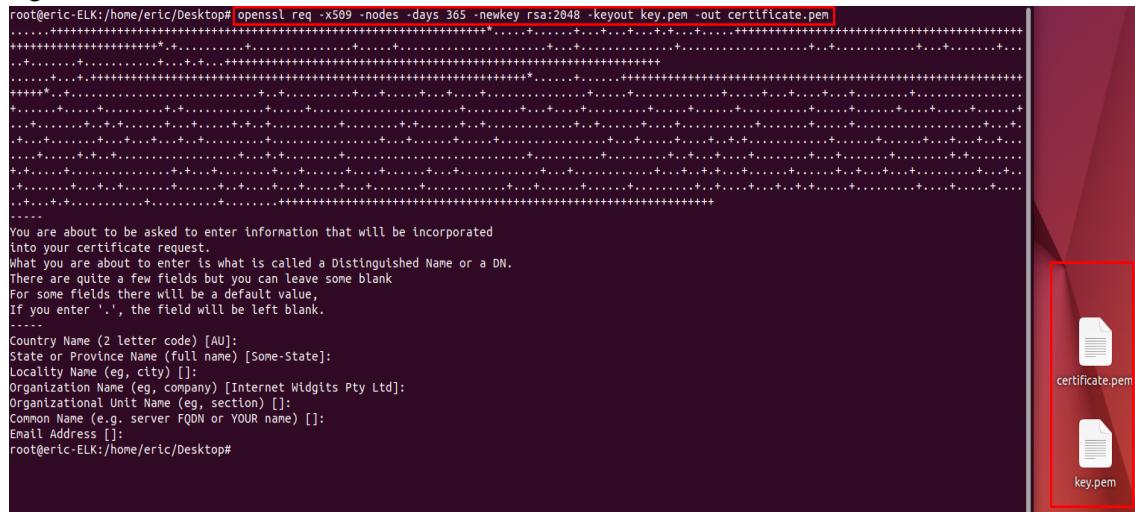
Cancel Sign in

Si es exitoso entraremos en el **ElasticSearch**, si es fallido nos saltará el siguiente error



HTTPS con ElasticSearch

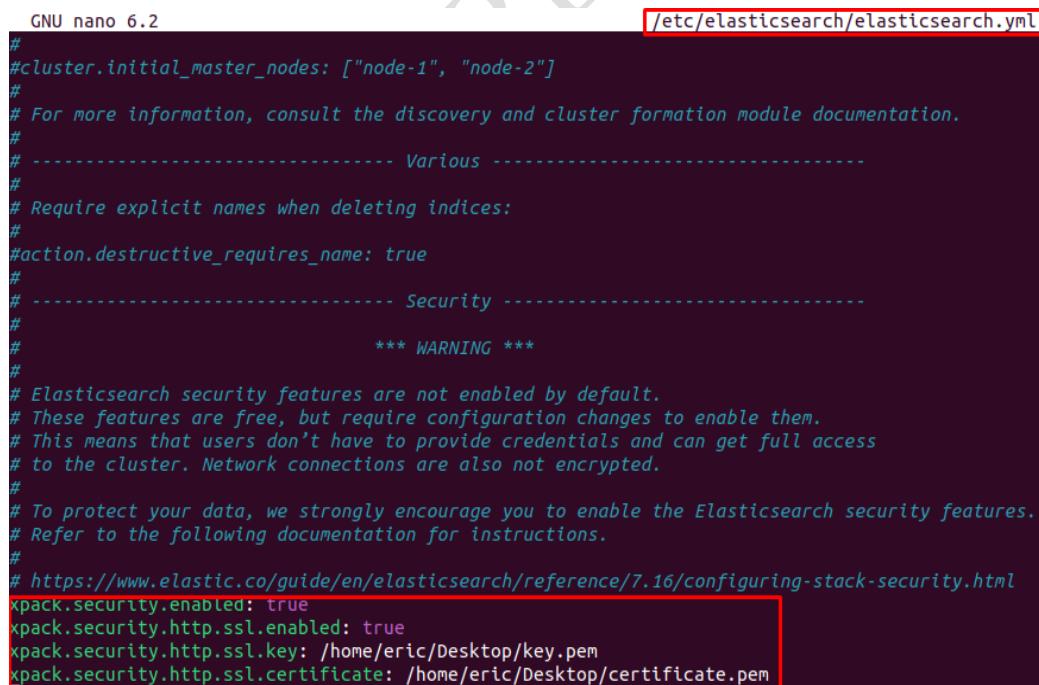
Para cifrar las comunicaciones de los clientes **HTTP** (navegadores o herramientas de consola) con **Elasticsearch** y **Kibana** utilizando **HTTPS** y certificados **TLS/SSL**, haremos lo siguiente: Generaremos una clave privada (**key.pem**) y el certificado autofirmado (**certificate.pem**) con el siguiente comando:



```
root@eric-ELK:/home/eric/Desktop# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout key.pem -out certificate.pem
.....
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:
root@eric-ELK:/home/eric/Desktop#
```

Esto nos generará la clave y el autofirmado durante 365 días.

Lo siguiente será configurar **Elasticsearch** para habilitar **HTTPS** y utilizar el certificado **SSL/TLS**. Abrimos el archivo de configuración **elasticsearch.yml** y agregamos las siguientes líneas:



```
GNU nano 6.2
/etc/elasticsearch/elasticsearch.yml
#
#cluster.initial_master_nodes: ["node-1", "node-2"]
#
# For more information, consult the discovery and cluster formation module documentation.
#
# ----- Various -----
#
# Require explicit names when deleting indices:
#
#action.destructive_requires_name: true
#
# ----- Security -----
#
# *** WARNING ***
#
# Elasticsearch security features are not enabled by default.
# These features are free, but require configuration changes to enable them.
# This means that users don't have to provide credentials and can get full access
# to the cluster. Network connections are also not encrypted.
#
# To protect your data, we strongly encourage you to enable the Elasticsearch security features.
# Refer to the following documentation for instructions.
#
# https://www.elastic.co/guide/en/elasticsearch/reference/7.16/configuring-stack-security.html
xpack.security.enabled: true
xpack.security.http.ssl.enabled: true
xpack.security.http.ssl.key: /home/eric/Desktop/key.pem
xpack.security.http.ssl.certificate: /home/eric/Desktop/certificate.pem
```

Hacemos lo mismo pero con la configuración de **Kibana** de la siguiente manera:

```
GNU nano 6.2
/etc/kibana/kibana.yml

#elasticsearch.logQueries: false

# Specifies the path where Kibana creates the process ID file.
#pid.file: /run/kibana/kibana.pid

# Enables you to specify a file where Kibana stores log output.
#logging.dest: stdout

# Set the value of this setting to true to suppress all logging output.
#logging.silent: false

# Set the value of this setting to true to suppress all logging output other than error messages.
#logging.quiet: false

# Set the value of this setting to true to log all events, including system usage information
# and all requests.
#logging.verbose: false

# Set the interval in milliseconds to sample system and process performance
# metrics. Minimum is 100ms. Defaults to 5000.
#ops.interval: 5000

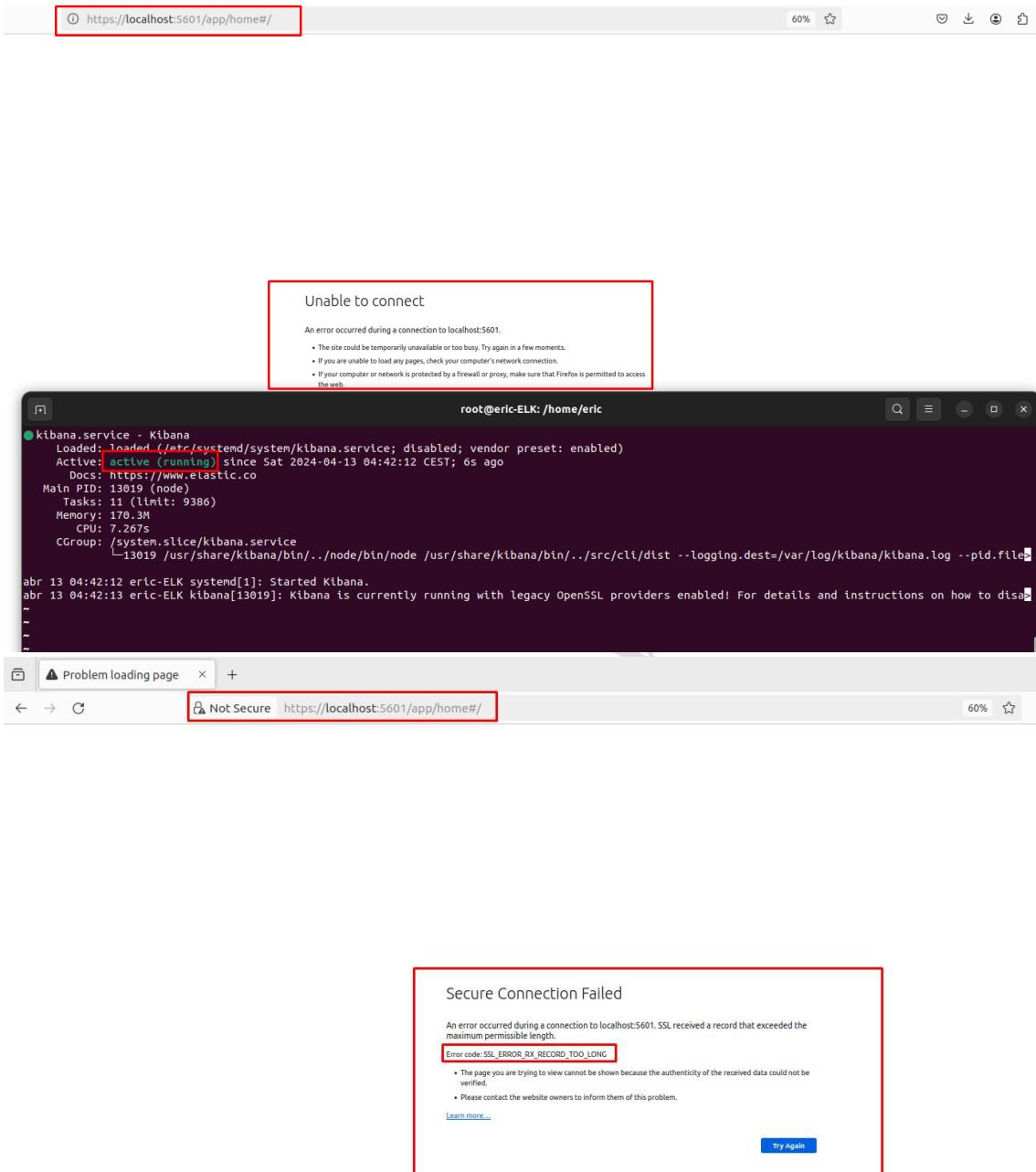
# Specifies locale to be used for all localizable strings, dates and number formats.
# Supported languages are the following: English - en , by default , Chinese - zh-CN .
#i18n.locale: "en"
server.ssl.enabled: true
server.ssl.certificate: /home/eric/Desktop/certificate.pem
server.ssl.key: /home/eric/Desktop/key.pem
```

Tras añadir ambas líneas a los servicios, debemos resetearlos con **systemctl restart "elasticsearch-kibana"**

```
root@eric-ELK:/var/lib/elasticsearch/nodes/0# systemctl restart elasticsearch
root@eric-ELK:/var/lib/elasticsearch/nodes/0# systemctl restart kibana
root@eric-ELK:/var/lib/elasticsearch/nodes/0#
```

SIEM con ELK – Eric Suárez Vázquez – Incidentes de ciberseguridad

Pese a hacer un status de ambos servicios y tener todo correctamente levantado, **HTTPS** en una máquina local con certificados autogenerados no es funcional, o supone muchos problemas para la máquina, sobretodo de rendimiento

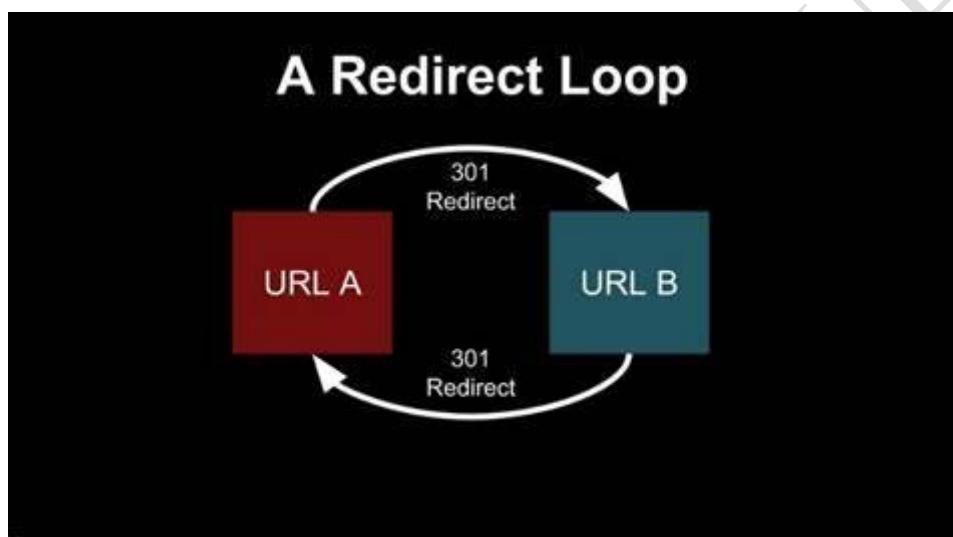


Atención: Agregar **HTTPS** en un entorno **localhost** puede introducir complejidad adicional y tener un impacto en el **rendimiento** y la **experiencia** de desarrollo.

Aunque puede ser útil en algunos casos para replicar fielmente el entorno de producción, también puede ser innecesario y **complicar innecesariamente** el proceso de desarrollo y pruebas.

En este caso tras realizar algunas pruebas y comprobar registros **Logs** de la propia herramienta de **Kibana**, esta para su función al ponerse en bucle infinito al encontrar la **key** y el **certificate.pem**, por ello no es capaz de cargar nuestro servicio ELK y no es posible cifrar las comunicaciones.

Esto **probablemente** sería diferente si **Servidor** y **Cliente** fueran distintas máquinas, ya que no entraría en bucle (eso sí, la máquina cliente debe también poseer el **certificate.pem**) para poder entrar al servicio por **HTTPS**.



Final

En resumen, **ELK** ofrece una solución **integral y potente** para la **gestión y análisis de datos** de **registros** y métricas, proporcionando a las organizaciones las herramientas necesarias para extraer **insights** valiosos de sus datos y mejorar la eficiencia operativa, la toma de decisiones y la experiencia del usuario, así como proteger la máquina de posibles ataques o errores del sistema.