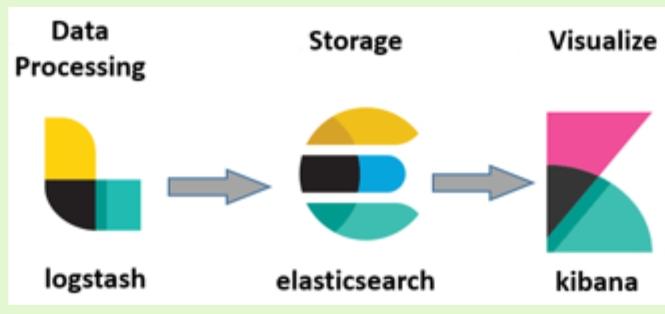


SIEM ELK.

Gestión de Eventos e Información de Seguridad



[Elastic. ELK Stack \(CC0\)](#)

La detección IDS es el primer estadio de tratamiento de la información en un SOC. La información recolectada en esta etapa deberá ser filtrada, clasificada y almacenada en las siguientes etapas de tratamiento, mediante las herramientas adecuadas.

Hecho esto, dará comienzo la etapa final, que se desarrollará de forma continua una vez iniciada, esto es, el análisis de información para detectar patrones de ataque y sacar las conclusiones correspondientes de cara a la Prevención de Incidentes de Ciberseguridad.

Esta unidad complementa las tareas efectuadas en la unidad anterior, esto es, Detección IDS, por lo que se partirá con Snort instalado, configurado y funcionando en la máquina del laboratorio.

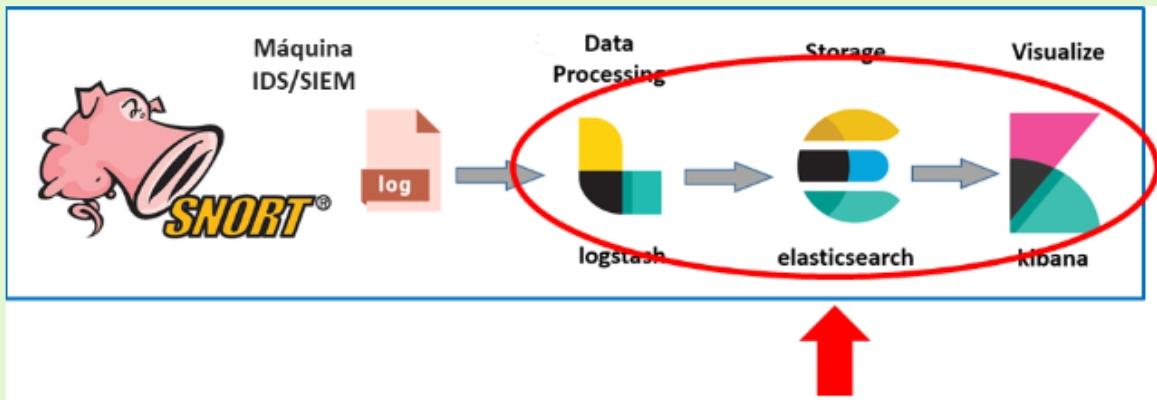
Así pues, se añadirán a la máquina el resto de componentes necesarios para disponer de un SIEM completo y operativo (se recomienda utilizar como mínimo una Raspberry Pi 4B 4GB como escenario de trabajo).

1.- Escenario de Trabajo SIEM.

Caso práctico

Para implementar el SIEM se instalarán en Raspbian los componentes Linux del Stack ELK, y se visualizará en un PC el dashboard para monitorización de información y eventos de seguridad. No será preciso instalar ningún SW en el PC, sólo hará falta disponer de un navegador.

El sistema resultante permitirá efectuar las labores de Detección de Intrusos, Prevención de Ataques, monitorización de intrusiones, preparación de Cuadros de Mando, Gestión de Alarmas, etc., que son las que se efectúan habitualmente en un SOC.



[Francisco Artés - Elaboración Propia. ELK Stack en el SOC \(CC0\)](#)

Esta unidad conecta con las tareas efectuadas en la unidad anterior, esto es, Detección IDS, por lo que se partirá con Snort instalado, configurado y funcionando en la máquina DMZ1.

En esta sesión se añadirán a la máquina el resto de componentes necesarios para disponer de un SIEM completo y operativo (se recomienda utilizar como mínimo una Raspberry Pi 4B 4GB como escenario de trabajo).

Para ello, se instalarán en Raspbian los componentes Linux del Stack ELK, y se visualizará en un PC el dashboard para monitorización de información y eventos de seguridad (no se precisa instalar ningún SW en el PC, sólo hará falta disponer de un navegador).

En resumidas cuentas, el sistema resultante permitirá efectuar las labores de Detección de Intrusos, Prevención de Ataques, monitorización de intrusiones, preparación de Cuadros de Mando, Gestión de Alarmas, etc., que son las que se efectúan habitualmente en un SOC.

1.1.- Premisas para la Práctica SIEM.

Premisas de Partida – Snort instalado y operativo

En primer lugar y para evitar problemas durante la instalación del SIEM, nos debemos asegurar de que el fichero de alertas de Snort tenga todos los permisos:

```
sudo chmod 777 /var/log/snort_alerts.log
```

En la sesión anterior ya incluimos las reglas de detección de ICMP y TCP para levantar alertas de acceso ping y ssh, por lo que pudimos comprobar que el fichero de alertas va creciendo adecuadamente según la progresión de los ataques.

Antes de proseguir con la instalación del SW del SIEM, es conveniente comprobar también que se actualiza la fecha y hora del fichero de log interno binario de Snort:

```
ls -l /var/log/snort/snort.log
```

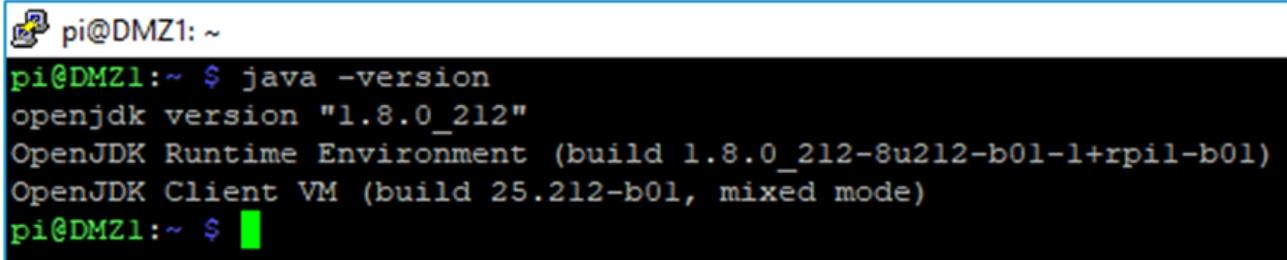
Efectuadas estas comprobaciones, detenemos Snort antes de proseguir con la instalación del SIEM:

```
sudo systemctl stop snort
```

1.2.- Instalación de OpenJDK.

Para comenzar, el stack ELK requiere la instalación del Open Java Development Kit, al menos en su versión 8, junto con un par de librerías adicionales (no obstante, se deberá chequear la versión del OpenJDK requerida por la versión de los productos de ELK que se vayan a instalar en cada caso):

```
sudo apt install openjdk-8-jdk libjffi-java libjffi-jni
```



```
pi@DMZ1: ~
pi@DMZ1: ~ $ java -version
openjdk version "1.8.0_212"
OpenJDK Runtime Environment (build 1.8.0_212-8u212-b01-1+rpi1-b01)
OpenJDK Client VM (build 25.212-b01, mixed mode)
pi@DMZ1: ~ $
```

Francisco Artés - Elaboración Propia. Captura de Pantalla con la Comprobación de Versión de JAVA ([CC0](#))

NOTA IMPORTANTE: Al ejecutar los comandos y transcribir los textos indicados a los ficheros correspondientes, deberemos poner especial atención en las comillas, como ya comentamos en la sesión anterior. Deberán figurar en todos los lugares indicados y no deberán ser las comillas tipográficas que en ocasiones insertan las aplicaciones de edición de textos.

1.3.- Instalación de Elasticsearch.

Elasticsearch es el corazón del SIEM.

Se trata de una base de datos tipo NoSQL, esto es, Not Only SQL. Estas bases de datos están preparadas para almacenar cualquier tipo de información de forma inmediata, aunque su formato no cuadre exactamente con la estructura interna. Esta flexibilidad unida a su capacidad para manejar grandes cantidades de datos, hacen que estos gestores de bases de datos sean muy apropiados para el ámbito de Big Data.

En el caso del SIEM su misión será almacenar toda la información relacionada con los incidentes detectados, para poder analizarla adecuadamente y extraer rápidamente las conclusiones necesarias de cara a la prevención.

1.3.1.- Descarga y Edición del Fichero de Configuración.

Descargamos el paquete Elasticsearch, que indexará y almacenará en su base de datos nuestras alertas de seguridad y nuestros logs. Lo haremos con wget y lo instalaremos con dpkg:

```
sudo wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.5.4.deb
```

```
sudo dpkg -i elasticsearch-6.5.4.deb
```

Editamos el fichero de configuración de elasticsearch y asignamos los siguientes valores a las variables:

```
sudo nano /etc/elasticsearch/elasticsearch.yml
```

```
cluster.name: siem.sky.net
```

```
node.name: node01.siem.sky.net
```

```
network.host: 192.168.1.21
```

```
discovery.type: single-node
```

```
xpack.ml.enabled: false
```

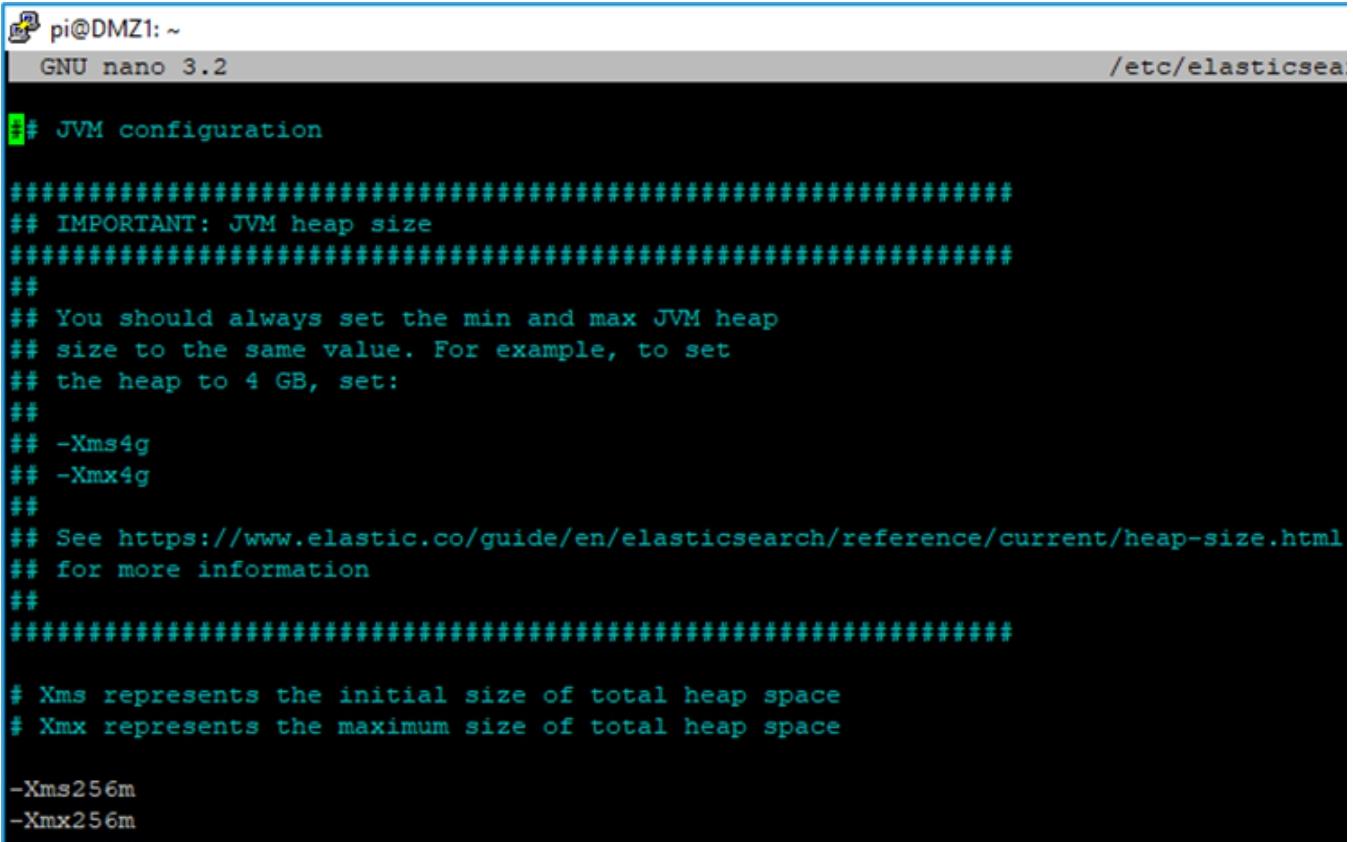
1.3.2.- Limitación del Uso de la Memoria RAM.

Configuramos elasticsearch para que utilice un máximo de 256 MB de RAM, lo cual resulta más que suficiente para un ejercicio de laboratorio:

```
sudo nano /etc/elasticsearch/jvm.options
```

```
-Xms256m
```

```
-Xmx256m
```



The screenshot shows a terminal window titled 'pi@DMZ1: ~' running 'GNU nano 3.2'. The file being edited is '/etc/elasticsearch/jvm.options'. The content of the file is as follows:

```
# JVM configuration
#####
## IMPORTANT: JVM heap size
#####
## You should always set the min and max JVM heap
## size to the same value. For example, to set
## the heap to 4 GB, set:
##
## -Xms4g
## -Xmx4g
##
## See https://www.elastic.co/guide/en/elasticsearch/reference/current/heap-size.html
## for more information
##
#####

# Xms represents the initial size of total heap space
# Xmx represents the maximum size of total heap space

-Xms256m
-Xmx256m
```

1.3.3.- Arranque y Chequeo de Status.

Arrancamos Elasticsearch y comprobamos su estado:

```
sudo systemctl start elasticsearch
```

```
sudo systemctl status elasticsearch
```

```
pi@DMZ1: ~
pi@DMZ1:~ $ sudo systemctl start elasticsearch
pi@DMZ1:~ $ sudo systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
   Loaded: loaded (/lib/systemd/system/elasticsearch.service; disabled; vendor preset: enabled)
   Active: active (running) since Thu 2021-09-09 16:18:51 CEST; 7s ago
     Docs: http://www.elastic.co
 Main PID: 14073 (java)
    Tasks: 15 (limit: 4915)
   CGroup: /system.slice/elasticsearch.service
           └─14073 /bin/java -Xms256m -Xmx256m -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupan

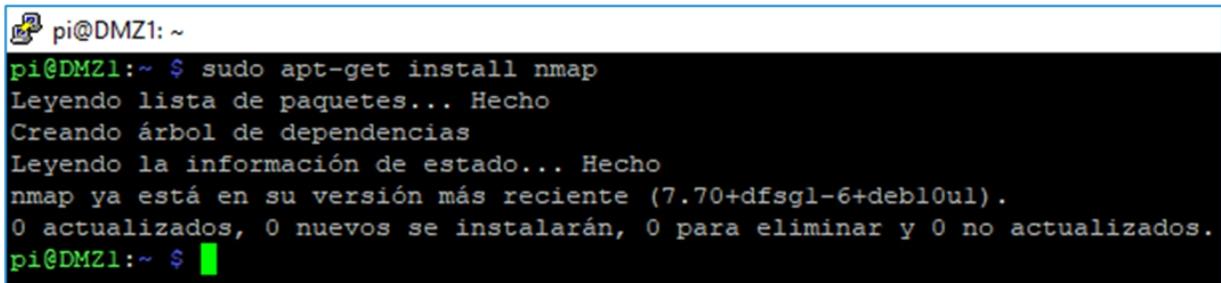
sep 09 16:18:51 DMZ1 systemd[1]: Started Elasticsearch.
pi@DMZ1:~ $
```

[Francisco Artés - Elaboración Propia. Captura de Pantalla con el Arranque y Status de Elasticsearch \(CC0\)](#)

1.3.4.- Uso de Nmap para Comprobar Acceso a Elasticsearch.

Finalmente, instalamos nmap para chequear si Elasticsearch está a la escucha en el puerto adecuado:

```
sudo apt-get install nmap
```



```
pi@DMZ1: ~
pi@DMZ1: ~ $ sudo apt-get install nmap
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
nmap ya está en su versión más reciente (7.70+dfsg1-6+deb10ul).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
pi@DMZ1: ~ $
```

[Francisco Artés - Elaboración Propia. Captura de Pantalla con la Instalación de nmap \(CC0\)](#)

Si todo está correctamente instalado y la aplicación está corriendo, Elasticsearch deberá estar a la escucha en el puerto 9200:

```
sudo nmap 192.168.1.21 -p9200
```

Además se podrá comprobar la operatividad de elasticsearch desde el PC, utilizando la dirección IP y el puerto 9200 en cualquier navegador:

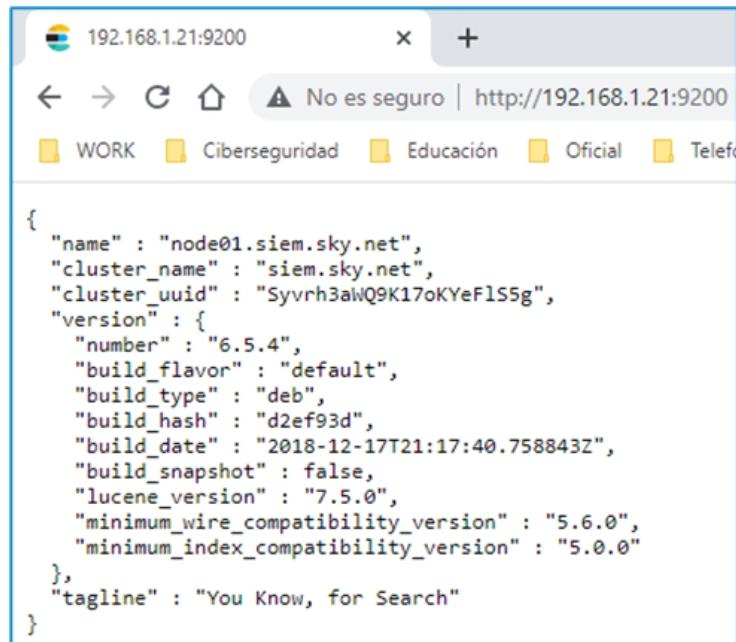
<http://192.168.1.21:9200>

```
pi@DMZ1: ~
pi@DMZ1:~ $ sudo nmap 192.168.1.21 -p9200
Starting Nmap 7.70 ( https://nmap.org ) at 2021-09-09 16:29 CEST
Nmap scan report for DMZ1 (192.168.1.21)
Host is up (0.00011s latency).

PORT      STATE SERVICE
9200/tcp  open  wap-wsp

Nmap done: 1 IP address (1 host up) scanned in 0.50 seconds
pi@DMZ1:~ $
```

Francisco Artés - Elaboración Propia. Captura de Pantalla con la Comprobación de Elasticsearch mediante nmap (CC0)



A screenshot of a web browser window. The address bar shows the URL `http://192.168.1.21:9200`. The page content displays the following JSON configuration for an Elasticsearch node:

```
{
  "name" : "node01.siem.sky.net",
  "cluster_name" : "siem.sky.net",
  "cluster_uuid" : "Syvrh3aWQ9K17oKYeFlS5g",
  "version" : {
    "number" : "6.5.4",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "d2ef93d",
    "build_date" : "2018-12-17T21:17:40.758843Z",
    "build_snapshot" : false,
    "lucene_version" : "7.5.0",
    "minimum_wire_compatibility_version" : "5.6.0",
    "minimum_index_compatibility_version" : "5.0.0"
  },
  "tagline" : "You Know, for Search"
}
```

Francisco Artés - Elaboración Propia. Captura de Pantalla con la Comprobación de Elasticsearch mediante Navegador (CC0)

1.4.- Instalación de Logstash.

Logstash es una herramienta de tipo ETL, esto es, *Extract, Translate & Load*.

Su misión es filtrar los datos crudos procedentes del IDS e insertar la información seleccionada en la base de datos Elasticsearch.

Como se verá a continuación, esta misión se lleva a cabo mediante entidades denominadas *pipelines*.

1.4.1.- Descarga, Instalación y Limitación del Uso de la RAM.

Antes de pasar a la instalación de logstash, paramos elasticsearch:

```
sudo systemctl stop elasticsearch
```

Logstash es la aplicación que analizará gramaticalmente las alertas y los logs de Snort, filtrando y estructurando la información antes de insertarla en la base de datos de Elasticsearch.

Se instala de una forma similar a elasticsearch, limitando igualmente la RAM utilizada en función de la CPU de que se disponga (recomendados también 256 MB):

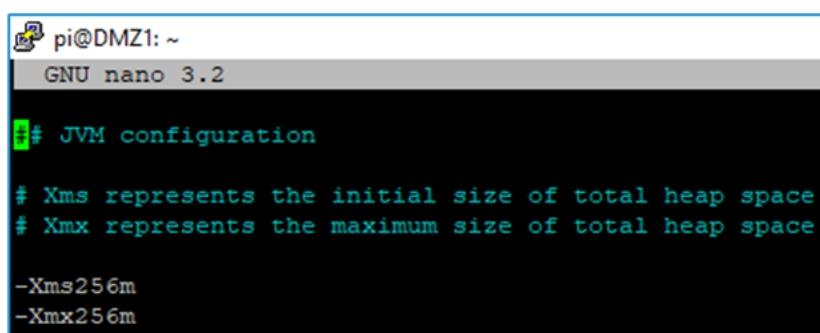
```
sudo wget https://artifacts.elastic.co/downloads/logstash/logstash-6.5.4.deb
```

```
sudo dpkg -i logstash-6.5.4.deb
```

```
sudo nano /etc/logstash/jvm.options
```

```
-Xms256m
```

```
-Xmx256m
```



A screenshot of a terminal window titled "pi@DMZ1: ~". The window shows the command "GNU nano 3.2" at the top. Below it, there is a file named "# JVM configuration". The content of the file is as follows:

```
# JVM configuration

# Xms represents the initial size of total heap space
# Xmx represents the maximum size of total heap space

-Xms256m
-Xmx256m
```

Francisco Artés - Elaboración Propia. Captura de Pantalla con el Ajuste de RAM en Logstash (CC0)

1.4.2.- Edición del Fichero de Configuración, Arranque y Comprobación del Status.

Editamos el fichero de configuración de logstash y asignamos los siguientes valores a las variables (atención a la dirección IP, que en este caso va entre comillas dobles):

```
sudo nano /etc/logstash/logstash.yml
```

```
node.name: node01.siem.sky.net
```

```
http.host: "192.168.1.21"
```

```
xpack.monitoring.enabled: false
```

```
xpack.management.enabled: false
```

Finalmente, arrancamos logstash, chequeamos su log de arranque para ver si todo ha ido bien, y lo detenemos antes de proseguir con la instalación de kibana:

```
sudo systemctl start logstash
```

```
sudo systemctl status logstash
```

1.4.3.- Revisión del Log de Arranque y Parada de la Aplicación.

Esperamos 2 minutos y chequeamos el log de arranque:

```
sudo tail -f /var/log/logstash/logstash-plain.log
```

Se observarán varios mensajes de error notificando que aún falta completar la configuración, no obstante, si aparece el mensaje "Successfully started Logstash API endpoint", esto indicará que todo ha ido bien.

Antes de proseguir, detenemos logstash y comprobamos que está parado:

```
sudo systemctl stop logstash
```

```
sudo systemctl status logstash
```

Deberá aparecer que está "inactive" o "failed"

1.5.- Instalación de Kibana.

En el stack ELK, Kibana es la herramienta que se utiliza para construir cuadros de mando interactivos (dashboards) y monitorizar las notificaciones y las alarmas.

Kibana 6.5.4 depende de Node.js 8.14. Esta versión de Node.js no es la que incluye de serie la instalación de Kibana. A continuación veremos cómo sustituirla.

IMPORTANTE: Kibana no puede convivir con otras aplicaciones que ocupen el puerto 80 en la misma máquina, por ejemplo, Apache. Si las hubiera, sería necesario desinstalarlas antes de proseguir con la instalación de Kibana.



[Elastic](#). Logotipo de Kibana (CC0)

1.5.1.- Descarga de Node.js.

Situarse en el directorio home del usuario sudoer (pi):

```
cd
```

Descargar la versión mencionada de Node.js siguiendo detalladamente los pasos que se indican a continuación.

Tras este procedimiento, el ejecutable quedará situado en el directorio correcto.

```
sudo wget https://nodejs.org/dist/v8.14.0/node-v8.14.0-linux-armv7l.tar.xz
```

```
sudo tar -xvf node-v8.14.0-linux-armv7l.tar.xz node-v8.14.0-linux-armv7l/bin/node
```

```
cd node-v8.14.0-linux-armv7l/bin
```

```
sudo cp ./node /usr/local/bin/
```

```
cd
```

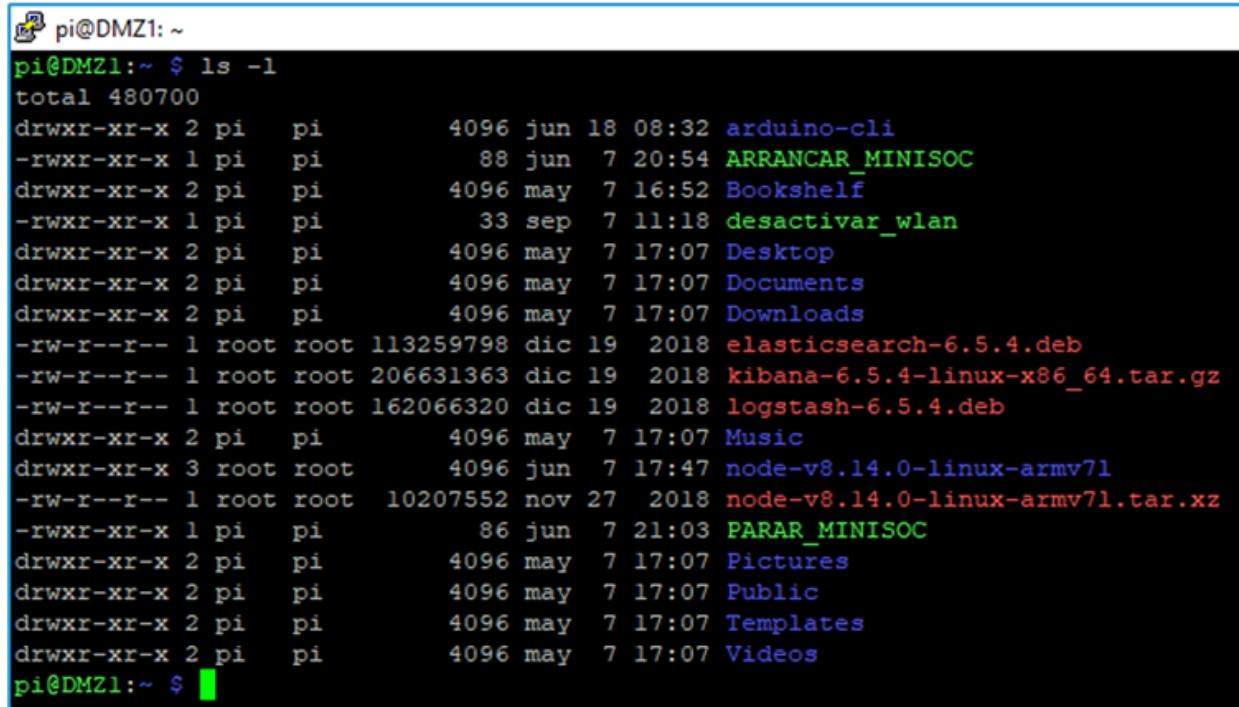
1.5.2.- Procedimiento de Instalación.

Instalar Kibana mediante el siguiente procedimiento:

```
sudo wget https://artifacts.elastic.co/downloads/kibana/kibana-6.5.4-linux-x86_64.tar.gz
```

```
sudo mkdir /usr/share/kibana/
```

```
sudo tar -xvf kibana-6.5.4-linux-x86_64.tar.gz --strip 1 --directory /usr/share/kibana/
```



```
pi@DMZ1:~ $ ls -l
total 480700
drwxr-xr-x 2 pi pi 4096 jun 18 08:32 arduino-cli
-rwxr-xr-x 1 pi pi 88 jun 7 20:54 ARRANCAR_MINISOC
drwxr-xr-x 2 pi pi 4096 may 7 16:52 Bookshelf
-rwxr-xr-x 1 pi pi 33 sep 7 11:18 desactivar_wlan
drwxr-xr-x 2 pi pi 4096 may 7 17:07 Desktop
drwxr-xr-x 2 pi pi 4096 may 7 17:07 Documents
drwxr-xr-x 2 pi pi 4096 may 7 17:07 Downloads
-rw-r--r-- 1 root root 113259798 dic 19 2018 elasticsearch-6.5.4.deb
-rw-r--r-- 1 root root 206631363 dic 19 2018 kibana-6.5.4-linux-x86_64.tar.gz
-rw-r--r-- 1 root root 162066320 dic 19 2018 logstash-6.5.4.deb
drwxr-xr-x 2 pi pi 4096 may 7 17:07 Music
drwxr-xr-x 3 root root 4096 jun 7 17:47 node-v8.14.0-linux-armv71
-rw-r--r-- 1 root root 10207552 nov 27 2018 node-v8.14.0-linux-armv71.tar.xz
-rwxr-xr-x 1 pi pi 86 jun 7 21:03 PARAR_MINISOC
drwxr-xr-x 2 pi pi 4096 may 7 17:07 Pictures
drwxr-xr-x 2 pi pi 4096 may 7 17:07 Public
drwxr-xr-x 2 pi pi 4096 may 7 17:07 Templates
drwxr-xr-x 2 pi pi 4096 may 7 17:07 Videos
pi@DMZ1:~ $
```

[Francisco Artés - Elaboración Propia](#). Captura de Pantalla con la Comprobación de la Descarga de Kibana (CC0)

1.5.3.- Configuración de Kibana.

Configurar Kibana editando el fichero siguiente:

```
sudo nano /usr/share/kibana/config/kibana.yml
```

Incluyendo las siguientes variables en su interior (no borrar las comillas):

```
server.port: 80
```

```
server.host: "192.168.1.21"
```

```
server.name: node01.siem.sky.net
```

```
elasticsearch.url: "http://192.168.1.21:9200"
```

```
logging.dest: /var/log/kibana.log
```

```
# The Kibana server's name. This is used for display purposes.  
server.name: node01.siem.sky.net  
  
# The URL of the Elasticsearch instance to use for all your queries.  
elasticsearch.url: "http://192.168.1.21:9200"
```

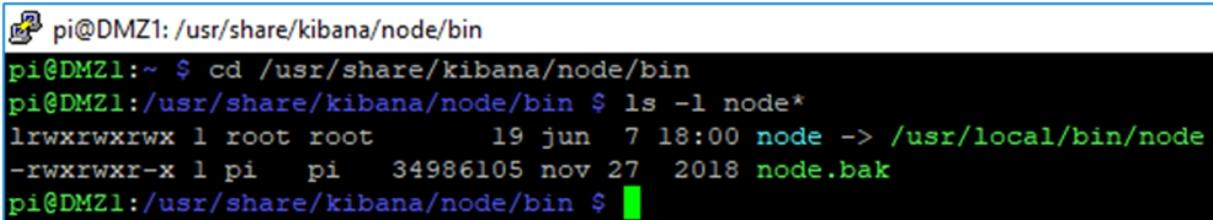
[Francisco Artés - Elaboración Propia](#). Captura de Pantalla con la Configuración de Kibana ([CC0](#))

1.5.4.- Sustitución de Node.js por su Versión Correcta.

Desactivar el Node.js que trae incluido Kibana, e indicarle que use la versión descargada ad hoc:

```
sudo mv /usr/share/kibana/node/bin/node /usr/share/kibana/node/bin/node.bak
```

```
sudo ln -s /usr/local/bin/node /usr/share/kibana/node/bin/node
```



```
pi@DMZ1: /usr/share/kibana/node/bin
pi@DMZ1:~ $ cd /usr/share/kibana/node/bin
pi@DMZ1:/usr/share/kibana/node/bin $ ls -l node*
lrwxrwxrwx 1 root root      19 jun  7 18:00 node -> /usr/local/bin/node
-rwxrwxr-x 1 pi   pi  34986105 nov 27  2018 node.bak
pi@DMZ1:/usr/share/kibana/node/bin $
```

Francisco Artés - Elaboración Propia. Captura de Pantalla con la Sustitución de Node (CC0)

1.5.5.- Edición de un Fichero de Servicio y Arranque.

Escribir un fichero de servicio "Systemd" para arrancar y parar Kibana:

```
sudo nano /etc/systemd/system/kibana.service
```

Incluir este contenido en el fichero de servicio:

```
[Unit]
```

```
Description=Kibana
```

```
[Service]
```

```
ExecStart=/usr/share/kibana/bin/kibana
```

```
[Install]
```

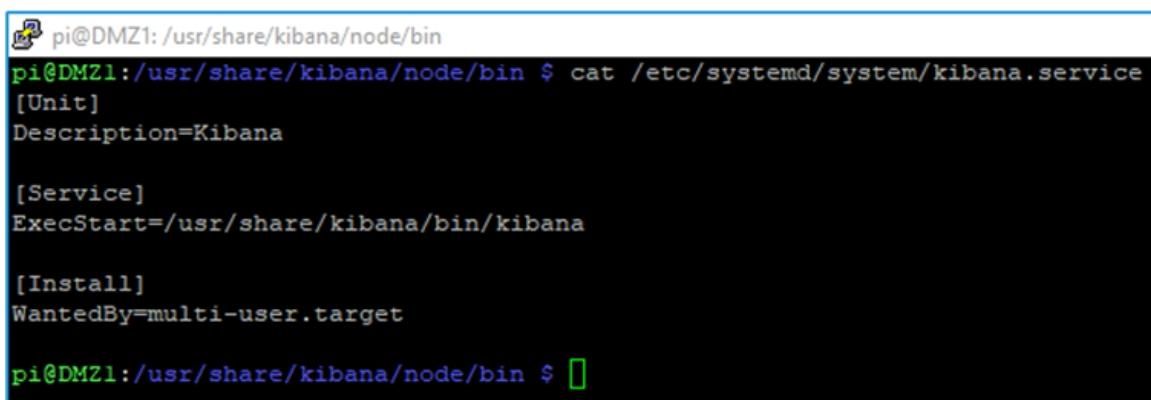
```
WantedBy=multi-user.target
```

Arrancar Kibana y comprobar su status:

```
sudo systemctl start kibana
```

```
sudo systemctl status kibana
```

Deberá aparecer que está "active (running)"



```
pi@DMZ1: /usr/share/kibana/node/bin
pi@DMZ1:/usr/share/kibana/node/bin $ cat /etc/systemd/system/kibana.service
[Unit]
Description=Kibana

[Service]
ExecStart=/usr/share/kibana/bin/kibana

[Install]
WantedBy=multi-user.target

pi@DMZ1:/usr/share/kibana/node/bin $
```

[Francisco Artés - Elaboración Propia.](#) Captura de Pantalla con el Fichero de Servicio de Kibana (CC0)

1.6.- Configuración SIEM.

Una vez que se han instalado todas las piezas de la torre ELK y se ha efectuado su configuración de bajo nivel, se puede dar paso al proceso de configuración funcional.

En él se verá que la configuración de Elasticsearch es prácticamente inexistente, mientras que Logstash requerirá definir pipelines y Kibana requerirá definir métricas y tableros.

1.6.1.- Arranque del SIEM.

Reiniciar el host:

```
sudo reboot
```

Arrancar Elasticsearch siempre ANTES que kibana:

```
sudo systemctl start elasticsearch
```

```
sudo systemctl status elasticsearch
```

Arrancar Kibana:

```
sudo systemctl start kibana
```

```
sudo systemctl status kibana
```

Comprobar que está arrancado Elasticsearch mediante el navegador del PC (devolverá su lista de características en formato JSON):

```
http://192.168.1.21:9200
```

Esperar 10 minutos antes de comprobar que está arrancado Kibana, introduciendo su URL en el navegador del PC (arrancará la Home de Kibana):

```
http://192.168.1.21
```

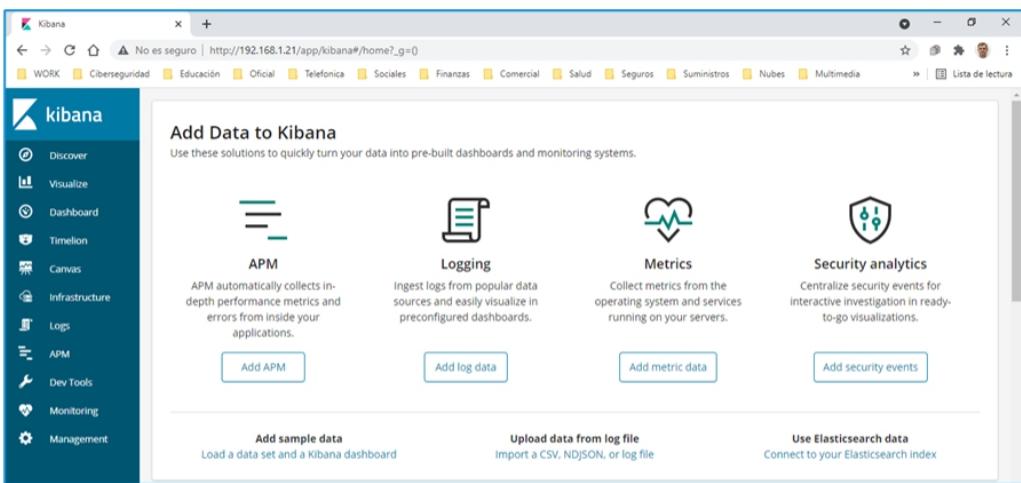
Arrancar el resto de las aplicaciones del SOC y comprobar que quedan activas:

```
sudo systemctl start logstash
```

```
sudo systemctl status logstash
```

```
sudo systemctl start snort
```

```
sudo systemctl status snort
```



[Francisco Artés - Elaboración Propia](#). Captura de Pantalla con la Home de Kibana (CC0)

1.6.2.- Pipelines en Logstash.

Configuración de logstash para capturar el log de Snort, procesar sus mensajes y grabarlos en Elasticsearch.

Para esto hay que definir una tubería de logstash (pipeline), con entrada, filtrado y salida hacia elasticsearch, en el fichero `/etc/logstash/conf.d/logstash.conf`

Las tuberías o pipelines tienen:

- ✓ **Entrada.** Indicación del punto de captura de datos, que puede ser un fichero local o remoto, un punto de acceso a un protocolo de comunicaciones, etc.
- ✓ **Filtrado.** La información de los logs y de los ficheros de trabajo está estructurada de forma muy variada, por lo que hay que utilizar la herramienta de filtrado Grok para formatearla antes de insertarla en la base de datos. Esta herramienta está incluida dentro de Logstash.
- ✓ **Salida.** Logstash puede producir salidas en varios formatos de ficheros, o bien, grabarla directamente en una base de datos NoSQL, como es el caso de Elasticsearch.

Entrada al Pipeline

Log de Snort que leerá Logstash:

```
/var/log/snort_alerts.log
```

Como resultado de la configuración efectuada en la sesión anterior, en el fichero /etc/snort/rules/local.rules tendremos las reglas siguientes:

```
# Regla para detectar un ping (ICMP)  
  
alert icmp any any -> $HOME_NET any (msg:"¡Trafico ICMP!"; sid:3000001;)  
  
#Regla para detectar un ssh (TCP y puerto 22)  
  
alert tcp any any -> any 22 (msg: "Acceso SSH"; sid:3000002;)
```

Editamos pues el fichero /etc/logstash/conf.d/logstash.conf.

Creamos la entrada a Logstash, para captura del fichero de log de Snort.

```
input {  
    file {  
        path => ["/var/log/snort_alerts.log"]  
        start_position => beginning  
    }  
}
```

Filtrado Grok en el Pipeline

Se efectuará el filtrado de la información entrante al pipeline con la potente herramienta Grok. Aunque la construcción de los patrones Grok se revisará a continuación con varios ejemplos, en este momento estudiaremos un patrón sencillo, con la sintaxis siguiente:

```

filter {

grok {

match => {"message" => "%{GREEDYDATA:cadena}"}

}

}

```

Este patrón es el más simple de todos, pues captura una línea completa del log de Snort, la asigna a variable “cadena” como un string, y la inserta en la base de datos de Elasticsearch.

Existe abundante documentación descriptiva de la sintaxis de los Grok Patterns.

Además, en Kibana hay un depurador Grok que permite ejecutar reglas de forma unitaria sobre una línea de log de muestra y comprobar si las reglas funcionan correctamente.

kibana -> Dev Tools -> Grok Debugger

The screenshot shows the Kibana Grok Debugger interface. At the top, under "Sample Data", there is a log entry:

```

1 Sep 29 17:37:38 claudiosiem snort: [1:3000002:0] Acceso SSH {TCP} 192.168.1.64:55317 -> 192.168.1.29:22
2

```

Below it, under "Grok Pattern", is the pattern definition:

```

1 %{GREEDYDATA:cadena}
2

```

Under "Structured Data", the output is shown as:

```

1 {
2   "cadena": "Sep 29 17:37:38 claudiosiem snort: [1:3000002:0] Acceso SSH {TCP} 192.168.1.64:55317 -> 192.168.1.29:22"
3 }

```

A "Simulate" button is visible below the pattern input field. At the bottom, there is a footer note:

[Francisco Artés - Elaboración Propia. Captura de Pantalla del Grok Debugger \(CC0\)](#)

Salida del Pipeline hacia Elasticsearch

Definición de la salida del pipeline, que en este caso será el índice “logstash” en la base de datos Elasticsearch:

```
output {  
  elasticsearch {  
    hosts => [ "http://192.168.1.21:9200" ]  
    index => "logstash"  
  }  
}
```

Comprobación del Funcionamiento del Pipeline

Para comprobar el funcionamiento del pipeline recién creado, lanzaremos un Ping y un ataque SSH desde otra máquina en sendos terminales, de forma que se registren los eventos asociados a dichos ataques.

Francisco Artés - Elaboración Propia. Captura de Pantalla con Detección de Tráfico SSH (CC0)

Desde kibana se podrá comprobar que Elasticsearch está ingiriendo la información formateada que le envía logstash a través de la tubería creada.

Para ello se deberá ir a Dev Tools, seleccionar Console y lanzar el comando siguiente para ver el contenido del índice logstash:

POST logstash/ search

The screenshot shows the Kibana interface with the sidebar menu open. The 'Dev Tools' option is selected. In the main area, the 'Console' tab is active, displaying a code editor with the following content:

```
1 POST logstash/_search
```

[Francisco Artés - Elaboración Propia.](#) Captura de Pantalla con el Comando POST (CC0)

Se obtendrá una respuesta similar a la de la figura adjunta.

The screenshot shows the Kibana interface with the sidebar menu open. The 'Dev Tools' option is selected. In the main area, the 'Console' tab is active, displaying a code editor with the following content:

```
1 POST logstash/_search
```

Below the code editor, the response is displayed as a JSON object with line numbers:

```
1 {  
2   "took" : 16,  
3   "timed_out" : false,  
4   "_shards" : {  
5     "total" : 5,  
6     "successful" : 5,  
7     "skipped" : 0,  
8     "failed" : 0  
9   },  
10  "hits" : {  
11    "total" : 1454347,  
12    "max_score" : 1.0,  
13    "hits" : [  
14      {  
15        "_index" : "logstash",  
16        "_type" : "doc",  
17        "_id" : "xbDqDXUBy2YtQahxgJ41",  
18        "_score" : 1.0,  
19        "_source" : {  
20          "@timestamp" : "2020-10-09T15:11:13.328Z",  
21          "path" : "/var/log/snort_alerts.log",  
22          "@version" : "1",  
23          "message" : "Oct 9 17:11:12 claudiosiem snort: [1:527:8] BAD  
-TRAFFIC same SRC/DST [Classification: Potentially Bad Traffic]  
[Priority: 2] [IPV6-ICMP] :: -> ff02::1:ff3b:4c89",  
24          "tags" : [  
25            "_grokparsefailure"  
26          ],  
27          "host" : "claudiosiem"  
28      }  
29    }  
30  }  
31 }
```

[Francisco Artés - Elaboración Propia.](#) Captura de Pantalla con el Resultado del Comando POST (CC0)

Patrón Grok Genérico para un String

```
Sep 29 17:37:38 claudiosiem snort: [1:3000002:0] Acceso SSH {TCP} 192.168.1.64:55317 -> 192.168.1.29:22
```

```
%{GREEDYDATA:cadena}
```

```
{
```

```
"cadena": "Sep 29 17:37:38 claudiosiem snort: [1:3000002:0] Acceso SSH {TCP} 192.168.1.64:55317 -> 192.168.1.29:22"
```

```
}
```

Nota. GREEDYDATA. Patrón “Codicioso” – No filtra nada. Captura TODO y lo guarda en una variable

Patrón Grok para Extraer Varios Campos

```
Sep 29 17:37:38 claudiosiem snort: [1:3000002:0] Acceso SSH {TCP} 192.168.1.64:55317 -> 192.168.1.29:22
```

```
%{MONTH:mes}%{SPACE}%{MONTHDAY:dia}%{SPACE}%{HOUR:hora}: %{MINUTE:minutos}: %{SECOND:segundos}%{GRE
```

```
{
```

```
    "hora": "17",
```

```
    "resto_mensaje": " claudiosiem snort: [1:3000002:0] Acceso SSH {TCP} 192.168.1.64:55317 -> 192.168.1.29:22",
```

```
    "segundos": "38",
```

```
    "mes": "Sep",
```

```
    "minutos": "37",
```

```
    "dia": "29"
```

```
}
```

Patrón Grok Literal para Extraer Todos los Campos

```
Sep 29 17:37:38 claudiosiem snort: [1:3000002:0] Acceso SSH {TCP} 192.168.1.64:55317 -> 192.168.1.29:22
```

```
%{MONTH:mes}%{SPACE}%{MONTHDAY:dia}%{SPACE}%{HOUR:hora}: %{MINUTE:minutos}: %{SECOND:segundos}%{SPA(
```

```
{ "accion": "Acceso", "nombre_maquina": "claudiosiem", "ip_destino": "192.168.1.29", "ip_origen": "192.168.1.64", "hora": "17", "m
```

Patrón con Comodines (Puntos)

```
Sep 29 17:37:38 claudiosiem snort: [1:3000002:0] Acceso SSH {TCP} 192.168.1.64:55317 -> 192.168.1.29:22
```

```
%{MONTH:mes}.%{MONTHDAY:dia}.%{HOUR:hora}.%{MINUTE:minutos}.%{SECOND:segundos}.%{WORD:nombre_maquina}
```

```
{ "accion": "Acceso", "nombre_maquina": "claudiosiem", "ip_destino": "192.168.1.29", "ip_origen": "192.168.1.64", "hora": "17", "m
```

Patrón Grok con Timestamp y Greedydata

```
Sep 29 17:37:38 claudiosiem snort: [1:3000002:0] Acceso SSH {TCP} 192.168.1.64:55317 -> 192.168.1.29:22
```

```
%{SYSLOGTIMESTAMP:fecha}.%{WORD:maquina}%{GREEDYDATA:mensaje}%{IP:dir_a}:%{INT:port_a}....%{IP:dir_b}:%{INT
```

```
{ "fecha": "Sep 29 17:37:38", "dir_b": "192.168.1.29", "port_a": "55317", "port_b": "22", "maquina": "claudiosiem", "mensaje": " snoi
```

Pipeline para Inyección de Snort a Kibana

Para continuar con nuestro laboratorio y diseñar un dashboard en Kibana, asignaremos el siguiente contenido al fichero /etc/logstash/conf.d/logstash.conf, antes de proseguir:

```
input {
  file {
    path => ["/var/log/snort_alerts.log"]
    start_position => beginning
  }
}

filter {
  grok {
    match => {"message" => "%{SYSLOGTIMESTAMP:fecha_y_hora}.%{WORD:maquina}.%{WORD:nombre_ids}\[%{INT:proceso}\]"}
    match => {"message" => "%{SYSLOGTIMESTAMP:fecha_y_hora}.%{WORD:maquina}.%{WORD:nombre_ids}\[%{INT:proceso}\]"}
  }
}

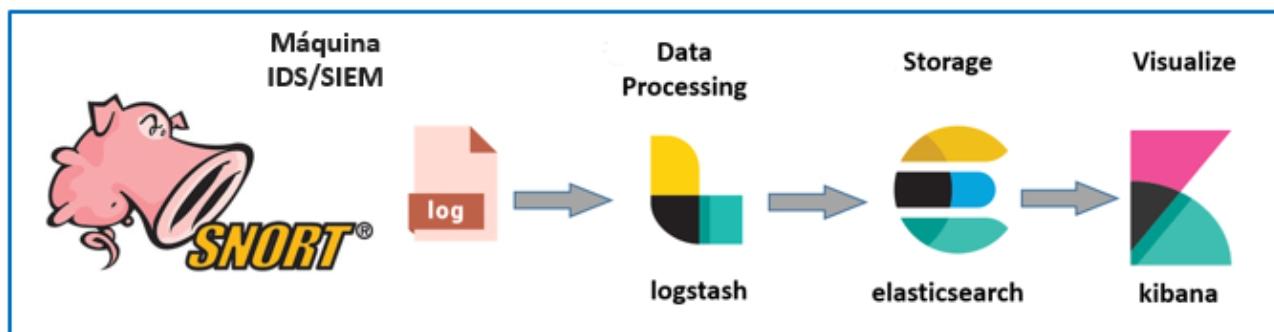
output {
  elasticsearch {
    hosts => [ "http://192.168.1.21:9200" ]
    index => "logstash"
  }
}
```

1.7.- Visualización SIEM.

A continuación monitorizaremos con Kibana la información que Logstash ha capturado de Snort, filtrado con Grok, e insertado en la base de datos de Elasticsearch.

Para ello, localizaremos la información en Elasticsearch a partir del índice creado por Logstash, añadiremos métricas basadas en dicho índice, y finalmente crearemos tableros basados en estas métricas.

Hecho esto, generaremos tráfico ICMP y SSH desde otra máquina para ver cómo se detectan dichos ataques y cómo se muestran en el tablero correspondiente.

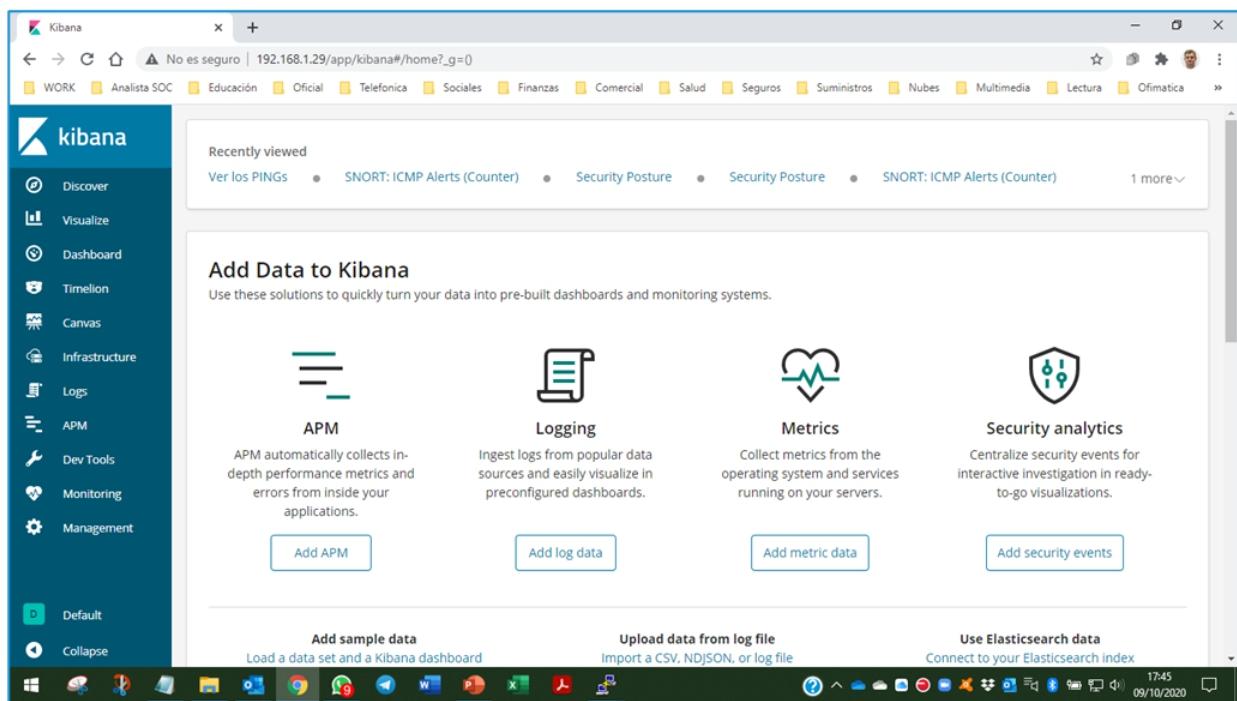


[Francisco Artés - Elaboración Propia. IDS y SIEM \(CC0\)](#)

1.7.1.- Arranque de Kibana desde el Navegador.

Arrancamos un navegador en el PC y escribimos la dirección de la máquina DMZ1 (sin especificar ningún puerto).

Si el stack ELK está correctamente arrancado, se visualizará la Home de Kibana.

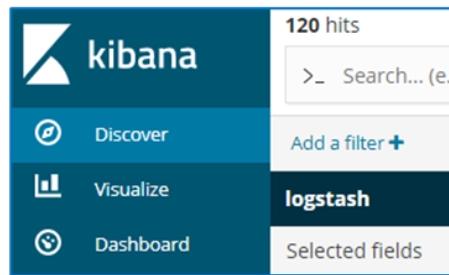


[Francisco Artés - Elaboración Propia](#). Captura de Pantalla con la Home de Kibana (CC0)

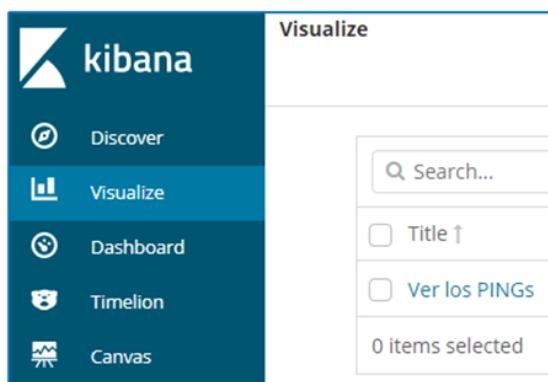
1.7.2.- Configuración de Kibana.

Para la configuración de Kibana, usaremos principalmente las siguientes secciones del menú de la página web:

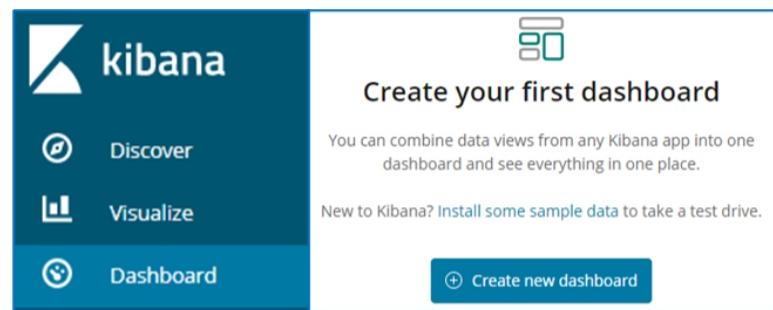
- ✓ **Discover.** En esta sección se localizan los índices creados en la base de datos. En nuestro caso, el índice que ha creado y que alimenta nuestro pipeline se denomina “logstash”.
- ✓ **Visualize.** Sección para la creación de métricas. Aquí definiremos métricas para Ping y para SSH. Programaremos además un refresco periódico (por ejemplo, 5 segundos), porque en caso contrario el tablero sólo efectuará una primera consulta al índice y se detendrá.
- ✓ **Dashboard.** Sección para diseño de Tableros o Cuadros de Mando, en función de las métricas definidas en el paso anterior.



[Francisco Artés - Elaboración Propia. Captura de Pantalla con Kibana Discover \(CC0\)](#)



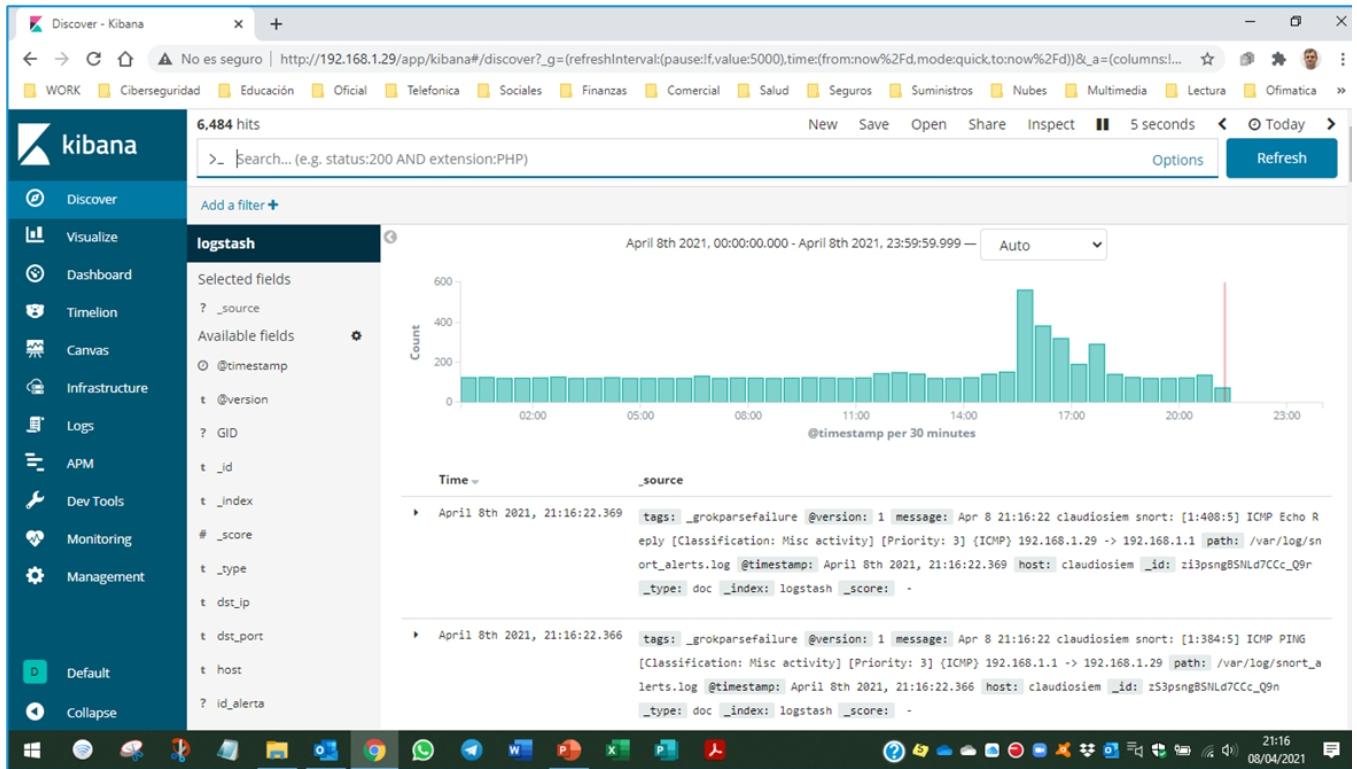
[Francisco Artés - Elaboración Propia. Captura de Pantalla con Kibana Visualize \(CC0\)](#)



[Francisco Artés - Elaboración Propia](#). *Captura de Pantalla con Kibana Dashboard (CC0)*

1.7.3.- Manejo de Kibana.

El índice “logstash” que hemos creado ya contiene la historia de todos los ataques efectuados desde el momento de su creación.



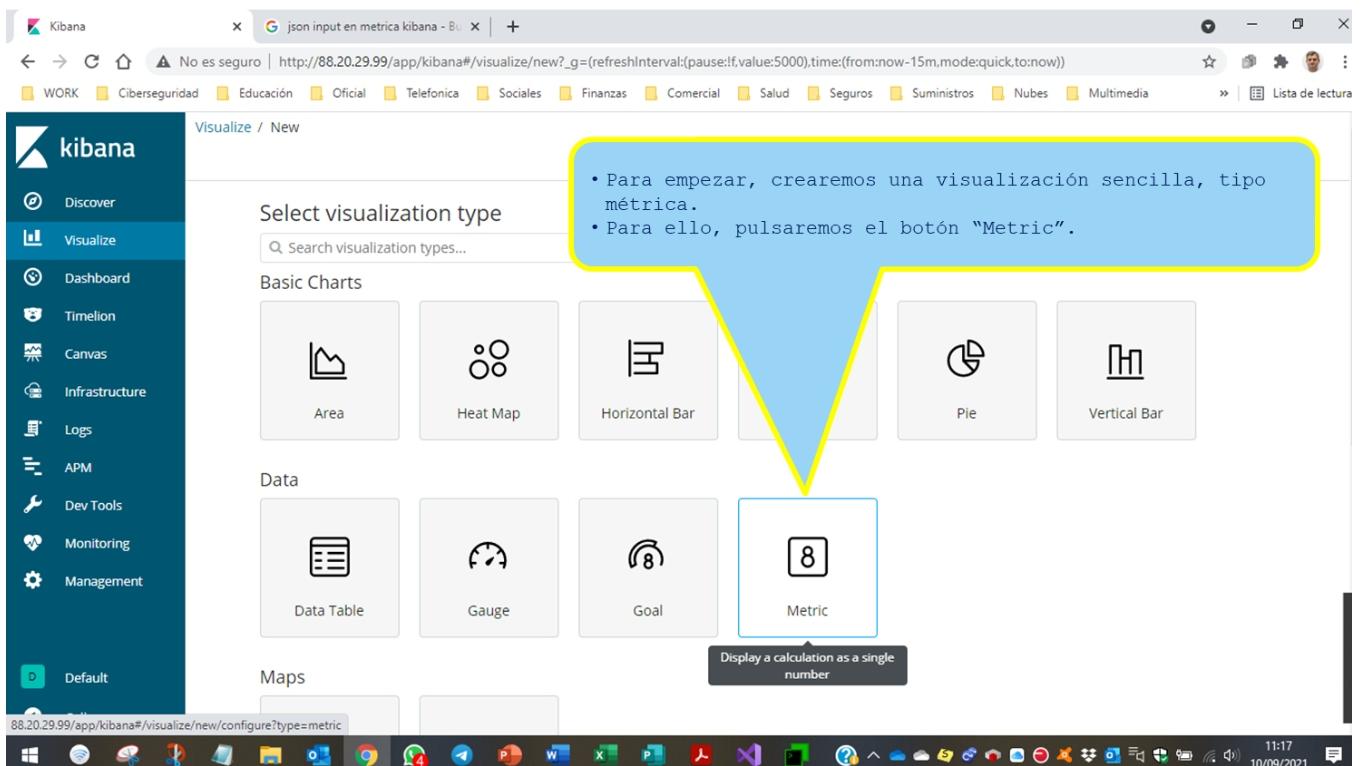
Francisco Artés - Elaboración Propia. Captura de Pantalla con el Contenido del Índice Logstash (CC0)

The screenshot shows the Kibana Discover interface. On the left, a sidebar lists various navigation options: Discover, Visualize, Dashboard, Timelion, Canvas, Infrastructure, Logs, APM, Dev Tools, Monitoring, Management, Default, and Collapse. The 'Discover' option is currently selected. The main area displays the 'logstash' index with 8,039 hits. A search bar at the top contains the placeholder 'Search... (e.g. status:200 AND extension:PHP)'. Below the search bar are buttons for 'New', 'Save', 'Open', 'Share', 'Inspect', and 'Refresh'. A yellow callout box highlights the 'Discover' option in the sidebar and the 'logstash' index in the main list. Another yellow callout box highlights the list of log entries, which include fields like '_source', 'puerto_a', 'message', 'dir_b', 'proceso', 'host', 'fecha_y_hora', 'maquina', 'mensaje', 'version', 'path', 'protocolo', 'dir_a', 'nombre_ids', and '@timestamp'. The bottom of the screen shows a Windows taskbar with various pinned icons.

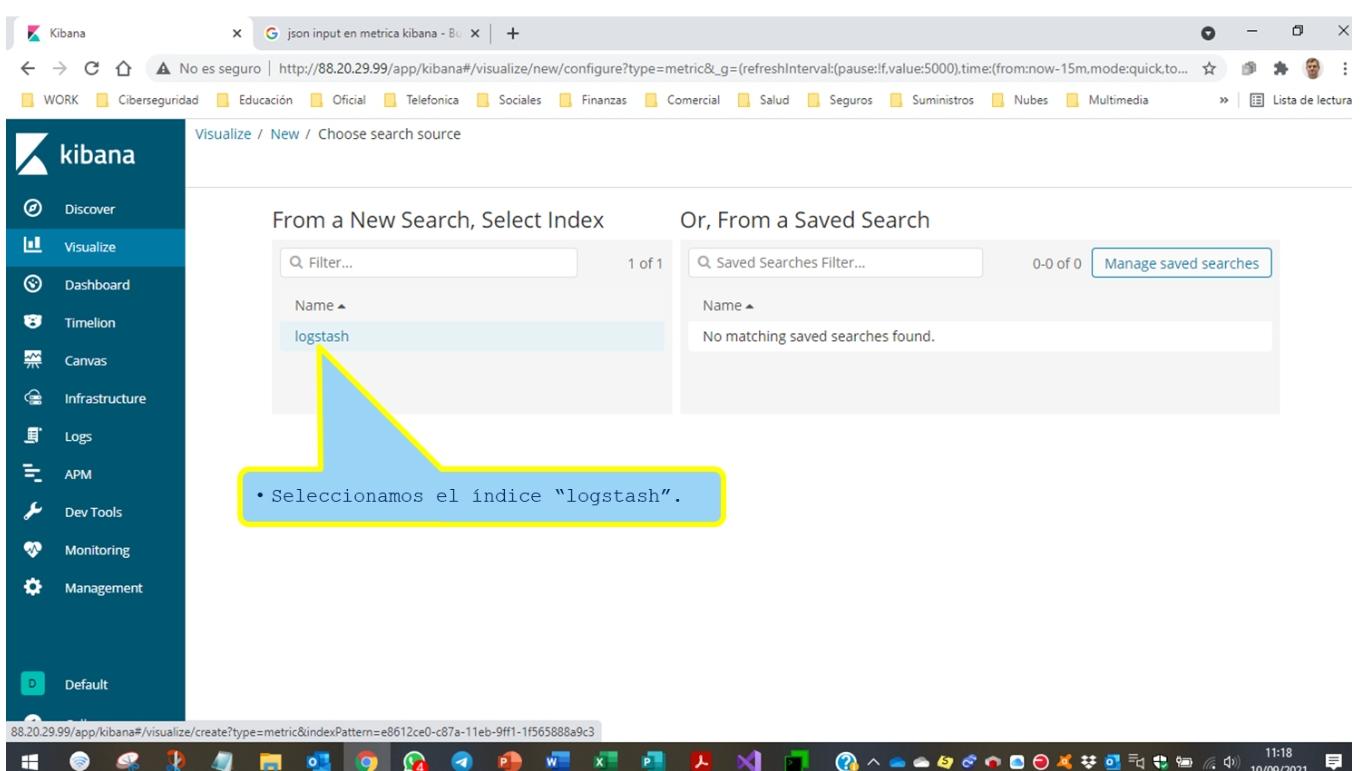
Francisco Artés - Elaboración Propia. Captura de Pantalla con el Discover del Índice Logstash (CC0)

The screenshot shows the Kibana Visualize interface. The left sidebar includes the same set of navigation options as the previous screenshot. The main area is titled 'Visualize' and features a search bar with the placeholder 'Search...'. Below the search bar is a button labeled '+ Create a visualization'. A yellow callout box highlights this button. The text 'Looks like you don't have any visualizations. Let's create some!' is displayed. At the bottom, it says '0 items selected'. The bottom of the screen shows a Windows taskbar with various pinned icons.

Francisco Artés - Elaboración Propia. Captura de Pantalla con la Creación de una Visualización (CC0)



Francisco Artés - Elaboración Propia. Captura de Pantalla con la Creación de una Métrica (CC0)



Francisco Artés - Elaboración Propia. Captura de Pantalla con la Selección de Índice (CC0)

The screenshot shows the Kibana interface with the left sidebar expanded. The 'Discover' and 'Visualize' buttons are selected. A modal window titled 'logstash' is open under the 'Metrics' tab, showing a single metric named 'Metric Count'. A large yellow callout bubble points from the text below to this metric. The main visualization area displays a large number '8,728' with the word 'Count' underneath.

- Creamos un contador, para lo cual seleccionaremos la opción "Count".
- Por el momento hay creado ningún filtro, por lo que se visualizará el recuento de todos los eventos registrados en el índice hasta el momento, independientemente de su naturaleza.

Francisco Artés - Elaboración Propia. Captura de Pantalla con la Creación de un Contador (CC0)

The screenshot shows the Kibana interface with the left sidebar expanded. The 'Discover' and 'Visualize' buttons are selected. A modal window titled 'logstash' is open under the 'Metrics' tab, showing a single metric named 'Metric'. The 'Custom Label' field contains the text 'Contador PINGs'. A large yellow callout bubble points from the text below to this label. The main visualization area displays a large number '8,973' with the word 'Count' underneath.

- Vamos a crear dos contadores, uno para ICMP y otro para eventos TCP, por lo que los denominaremos "Contador PINGs" y "Contador SSH".

Francisco Artés - Elaboración Propia. Captura de Pantalla con el Etiquetado del Contador (CC0)

• Para filtrar los eventos ICMP o TCP entre todos los eventos registrados, procedemos a crear un filtro.

• Para ello, pulsamos la opción "Add a filter".

Francisco Artés - Elaboración Propia. Captura de Pantalla con la Creación de un Filtro (CC0)

• En este caso usaremos parte del texto del mensaje para seleccionar los eventos, por lo que pulsaremos la opción "mensaje.keyword".

Francisco Artés - Elaboración Propia. Captura de Pantalla con la Selección de Eventos (CC0)

Kibana

json input en metrica kibana - Bu

No es seguro | http://

WORK Ciberseguridad Educación

kibana

Discover Visualize Dashboard Timeline Canvas Infrastructure Logs APM Dev Tools Monitoring Management

Default Collapse

Visualize / New > Search... (Add a filter + Add filter)

Filter mensaje.keyword is values... Edit Query DSL

Label Optional

Contador PINGS [1:3000002:0] Acceso SSH [1:3000001:0] Trafico ICMP! [1:384:5] ICMP PING [Classification: Misc activity] [Priority: 3] [1:408:5] ICMP Echo Reply [Classification: Misc activity] [Priority: 3] [1:366:7] ICMP PING *NIX [Classification: Misc activity] [Priority: 3]

Select metrics type Metric Cancel

Buckets Select buckets type

642 Count

A continuación, pulsaremos el operador "is". Al pulsar en el campo de contenidos, nos mostrará los diversos contenidos recibidos hasta el momento en el campo "Mensaje". Seleccionaremos el contenido correspondiente en el momento de crear cada uno de los dos contadores para detectar eventos ICMP/ping y TCP/SSH.

Francisco Artés - Elaboración Propia. Captura de Pantalla con la Selección de Contenidos (CC0)

The screenshot shows the Kibana Visualize interface. On the left, the sidebar includes options like Discover, Visualize (which is selected), Dashboard, Timelion, Canvas, Infrastructure, Logs, APM, Dev Tools, Monitoring, Management, Default, and Collapse. The main area displays a visualization titled "logstash". At the top, there's a search bar with the query "mensaje.keyword: \"[1:3000002:0] Acceso SSH\"". Below it, a button says "Add a filter +". A yellow callout box points to this button with the text: "• Hecho esto, quedará guardado el filtro en cuestión dentro de la visualización." The visualization itself shows a large number "4,422" with the label "Contador PINGs" underneath.

[Francisco Artés - Elaboración Propia](#). Captura de Pantalla con el Registro del Filtro dentro de la Visualización (CC0)

This screenshot is similar to the previous one but focuses on the "Refresh Interval" dropdown. The dropdown menu is open, showing options: Off, 5 seconds, 10 seconds, 1 minute, 5 minutes, 15 minutes, 30 minutes, 1 hour, 2 hours, 12 hours, and 1 day. The "5 seconds" option is highlighted. A yellow callout box points to this option with the text: "• Para que empiece a funcionar el filtro, tendremos que pulsar la opción "Refresh", o bien, activar un intervalo de refresco automático, por ejemplo, de 5 segundos." The visualization below shows a large number "4,475" with the label "Contador PINGs" underneath.

[Francisco Artés - Elaboración Propia](#). Captura de Pantalla con la Activación del Refresco (CC0)

The screenshot shows the Kibana interface with the 'Visualize' tab selected. A modal dialog titled 'logstash' is open, showing configuration for a metric visualization. The 'Metrics' section is set to 'Count' with a 'Custom Label' of 'Contador PINGs'. The 'Save' button is highlighted with a yellow arrow and a callout bubble containing the text: '• Pulsamos el botón "Save" para guardar los cambios efectuados.'

On the right, a large number '4,512' is displayed with the label 'Contador PINGs' below it.

• ATENCIÓN: Si el botón "Save" aparece inhabilitado, pulsar la flechita de "play" y se activará.

[Francisco Artés - Elaboración Propia. Captura de Pantalla con la Activación del Filtro y su Salvado \(CC0\)](#)

The screenshot shows the Kibana interface with the 'Visualize' tab selected. A modal dialog titled 'Save visualization' is open, prompting the user to enter a title for the saved visualization. The 'Title' field contains 'Contador SSH'. The 'Confirm Save' button is highlighted with a yellow arrow and a callout bubble containing the text: '• Asignamos un nombre a la visualización creada y pulsamos "Confirm Save".'

On the right, a large number '1,597' is displayed with the label 'Contador PINGs' below it.

[Francisco Artés - Elaboración Propia. Captura de Pantalla con el Etiquetado de la Visualización y su Salvado \(CC0\)](#)

1.7.4.- Creación de un Histograma.

- ✓ Pulsar "Visualize"
- ✓ Pulsar "+" para añadir una visualización
- ✓ Seleccionar "Vertical Bar"
- ✓ Pulsar "logstash**"
- ✓ Pulsar la flechita azul de "Y-Axis Count"
- ✓ Se abrirá un submenú, en el que seleccionaremos "Count" como "Aggregation" (~~generalmente seleccionado por defecto~~)
- ✓ Pondremos nombre al eje Y en el campo "Custom Label" (Número de PINGs / Número de SSHs)
- ✓ Pulsaremos "Add a Filter" y añadiremos el filtro de la forma habitual
- ✓ En el submenú "Buckets" pulsaremos X-Axis
- ✓ En "Aggregation" seleccionaremos la opción "Date Histogram", con Field="@timestamp" e intervalo automático.
- ✓ Tras lo anterior, pulsaremos la flechita azul de "play"
- ✓ Si el resultado es satisfactorio se pulsará "Save" en la barra de menú superior (se puede ver la gráfica interactiva si se ataca con ping o con ssh, en función del filtro elegido).

1.7.5.- Creación de un Tablero.

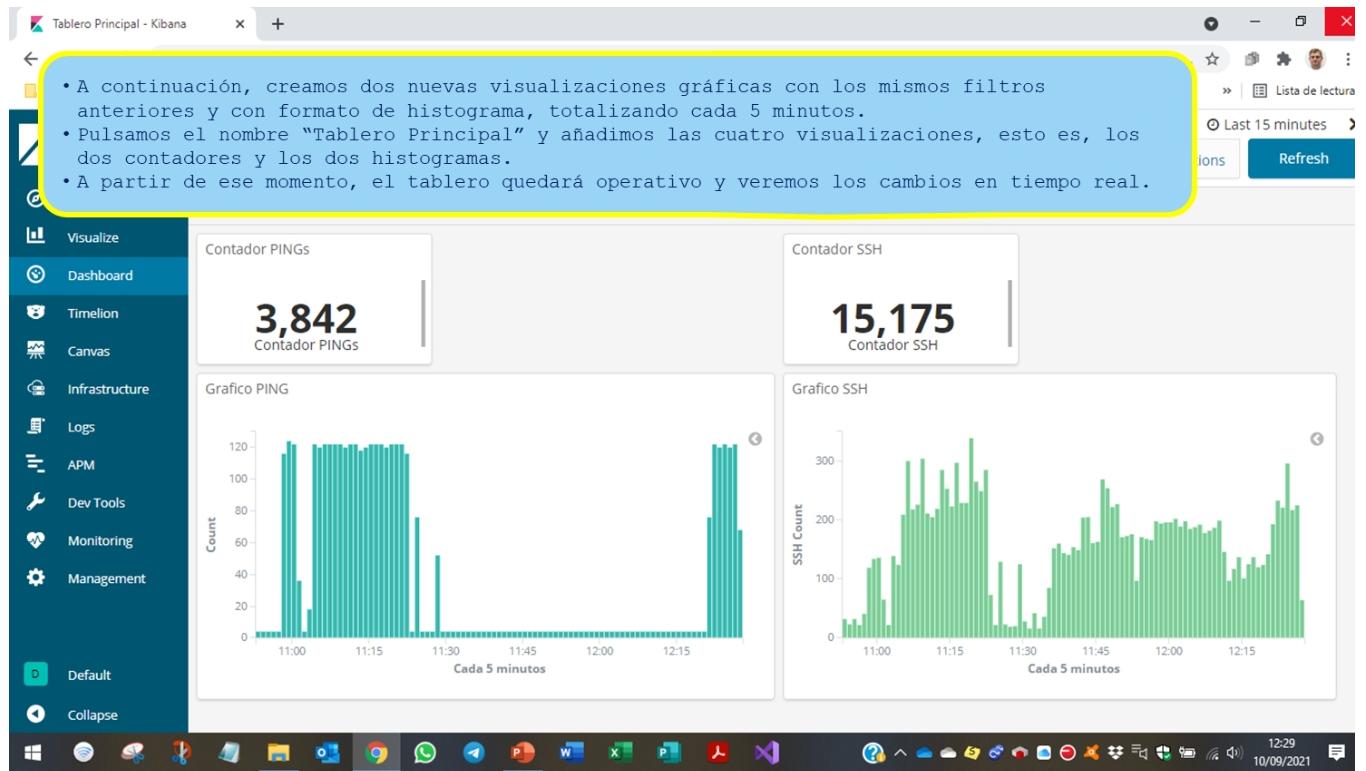
- ✓ Pulsar "Dashboard"
- ✓ Pulsar "Add" en la barra de menú superior para añadir visualizaciones al tablero
- ✓ Seleccionar secuencialmente todas las visualizaciones deseadas
- ✓ Cerrar el submenu
- ✓ Redimensionar y reposicionar las visualizaciones a gusto del usuario
- ✓ Pulsar "Save" y asignarle un nombre al tablero

The screenshot shows the Kibana interface. On the left is a sidebar with various navigation options: Discover, Visualize, Dashboard (which is selected), Timelion, Canvas, Infrastructure, Logs, APM, Dev Tools, Monitoring, Management, and Default. The main area is titled 'Dashboards' and contains a table with one row. The row has columns for 'Title' (with 'Tablero Principal' selected), 'Description', and 'Actions' (with a blue 'Edit' button). A yellow arrow points from the 'Edit' button to a callout box. The callout box contains the following text:

- A partir de este momento, las visualizaciones se podrán mostrar en pantalla de forma independiente, al seleccionarlas dentro de la opción "Visualize", no obstante, lo normal es crear un tablero de monitorización.
- Vamos a la opción "Dashboard", pulsamos el botón "Create new dashboard" y creamos un tablero denominado "Tablero Principal".

The browser address bar shows the URL: [http://88.20.29.99/app/kibana#/dashboards?_g=\(refreshInterval\(pause:if.value:5000\),time:\(from:now-15m.mode:quick,to:now\)\)](http://88.20.29.99/app/kibana#/dashboards?_g=(refreshInterval(pause:if.value:5000),time:(from:now-15m.mode:quick,to:now))). The taskbar at the bottom shows various open applications like File Explorer, Microsoft Edge, and others.

Francisco Artés - Elaboración Propia. Captura de Pantalla con la Creación de un Tablero Kibana (CCO)



2.- Bibliografía.

[Bibliografía](#) (pdf - 60141 B)

Condiciones y términos de uso de los materiales

Materiales desarrollados inicialmente por el Ministerio de Educación, Cultura y Deporte y actualizados por el profesorado de la Junta de Andalucía bajo licencia Creative Commons BY-NC-SA.



GOBIERNO
DE ESPAÑA
MINISTERIO
DE EDUCACIÓN
Y FORMACIÓN PROFESIONAL



Antes de cualquier uso leer detenidamente el siguiente [Aviso legal](#)

Historial de actualizaciones

Versión: 01.00.01

Fecha de actualización: 08/11/23

Actualización de materiales y correcciones menores.

Versión: 01.00.00

Fecha de actualización: 04/07/23

Versión inicial de los materiales.