

Instalación y configuración segura

DESCRIPCIÓN BREVE

Guía de instalación y configuración para establecer conexiones seguras entre los elementos del stack ELK. Guía para versiones desde la 8.X.

José Antonio Santos Gómez – CC0

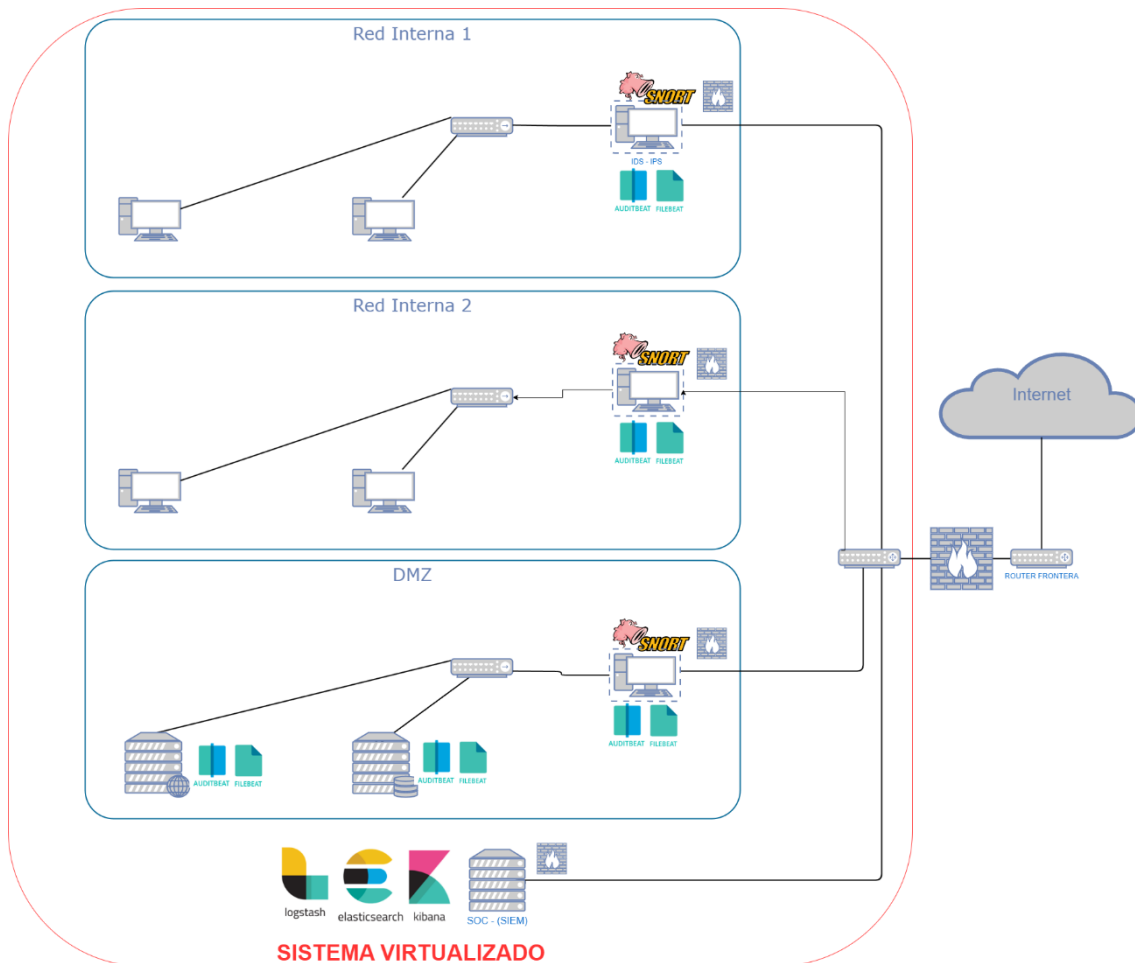
Incidentes de Ciberseguridad

Contenido

1	ESQUEMA DE RED DEL CASO PRÁCTICO:.....	2
2	COMUNICACIÓN ENTRE LAS MÁQUINAS DEL ESQUEMA:.....	3
2.1	Configuración de las máquinas virtuales para que tengan comunicación completa.	3
2.2	Configuración del IDS para registrar el tráfico de red. (SNORT)	5
3	ELASTIC STACK	6
3.1	Instalación de ElasticSearch.	6
3.2	Instalación y configuración de Kibana.	10
3.3	Instalación de FileBeats en los diferentes agentes IDS y principales servidores.	11
3.3.1	Configuración básica para salida por pantalla:.....	11
3.3.2	Habilitar un módulo en Filebeat:	13
3.3.3	Envío de datos a Logstash:	14
3.4	Instalación de Logstash.	14
3.4.1	Configuración de un pipeline:.....	16
3.4.2	Conexión de Logstash a Elasticsearch:.....	19
3.4.3	Creación de filtros en Logstash (filter)	22

1 ESQUEMA DE RED DEL CASO PRÁCTICO:

Dado el siguiente esquema de red:



En esta tarea se abordará la instalación de las herramientas: Elasticsearch, Kibana, Logstash (ELK) y Filebeat. Este conjunto de herramientas son las recomendadas por Elastic para la correcta implementación de un SIEM.

Las herramientas ELK serán instaladas en la máquina SOC (SIEM) para recoger los logs de los diferentes agentes IDS y datos de servidores. La herramienta Filebeat será instalada en los diferentes agentes que enviarán registros al SOC.

2 COMUNICACIÓN ENTRE LAS MÁQUINAS DEL ESQUEMA:

Esta parte es la realizada en la tarea anterior de la unidad 6. Aquí se muestra la configuración a realizar en las diferentes máquinas para conseguir la comunicación de todas ellas. En el caso de que la configuración ya se haya realizado con éxito, se puede avanzar hasta el siguiente punto.

2.1 Configuración de las máquinas virtuales para que tengan comunicación completa.

Pasos a realizar:

1. Configurar la máquina IDS (Snort) con dos interfaces de red:
 - a. La primera interfaz debe estar conectada a una red interna y será configurada de forma estática. La dirección a asignar a esta interfaz será la 192.168.10.1/24 ya que será la puerta de enlace predeterminada del servidor “webserver” ya proporcionado en la tarea. Esta interfaz debe tener una ruta estática que dirija el tráfico a la segunda interfaz.
 - b. La segunda interfaz debe estar conectada como adaptador puente y esta se puede configurar también de forma estática de modo que tenga una dirección válida de la red principal. A esta red es a la que pertenecen los diferentes equipos IDS y el SOC con el sistema SIEM, además estaría el sistema anfitrión y el router frontera. Esta interfaz debe tener la ruta por defecto para la salida por la puerta de enlace que posibilita la salida a Internet. (router de casa)

Ejemplo de configuración de red de la máquina IDS:

```
# This is the network config written by
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: no
      addresses: [192.168.1.140/24]
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
      routes:
        - to: default
          via: 192.168.1.1
    enp0s8:
      dhcp4: no
      addresses: [192.168.10.1/24]
      routes:
        - to: 192.168.1.0
          via: 192.168.1.140
```

2. Para que las máquinas de la red interna puedan salir correctamente hacia el exterior se deben realizar unas configuraciones adicionales en la máquina IDS:
 - a. Habilitar el bit de forward para que permita las comunicaciones entre las interfaces de red.

El uso de comandos como: “sudo sysctl -w net.ipv4.ip_forward=1” son temporales y se pierden al apagar la máquina. La manera de habilitar este bit de forma persistente sería editando el fichero “/etc/sysctl.conf” y agregar la siguiente línea:
net.ipv4.ip_forward = 1

Posteriormente se debe guardar y salir de la edición del fichero y aplicar la persistencia de este cambio con el comando: sudo sysctl -p

- b. Crear reglas de traducción de direcciones de origen (SNAT) y así los equipos puedan realizar las comunicaciones con el exterior.

Comandos necesarios:

- Acceder como root: `sudo su`
- Actualización de repositorios: `apt update`
- Instalación de iptables: `apt install iptables`
- Instalación de persistencia de iptables (Debian, Ubuntu):
`apt install iptables-persistent`
- Creación de regla de SNAT (según configuración de red):
`iptables -t nat -I POSTROUTING -o enp0s3 -j MASQUERADE`
- Guardado de la regla de forma persistente:
`iptables-save > /etc/iptables/rules.v4`

Como la política de las reglas del firewall por defecto son de lista negra (ACCEPT), no es necesario crear ninguna regla más en el firewall. Las reglas de permisión son necesarias en los firewalls con políticas de denegación (DROP, REJECT).

Desde este momento todas las máquinas de la red interna deberían tener conexión al exterior, ya sea a la red local compartida por los IDS como la salida a Internet. Los equipos de la red local (adaptadores puente) no podrían acceder a los equipos de las redes internas, ya que no tienen las rutas creadas para acceder a estas redes. Tenemos algunas opciones:

1. Crear rutas estáticas a nivel de equipo para acceder a estas redes internas. Este es una opción muy sencilla.
2. Crear rutas en la puerta de enlace (router de casa) para que todas las peticiones a estas redes sean redirigidas al equipo que puede manejarlas, es decir, al IDS que controla esa red interna. Si no se quiere tocar las rutas del router, lo mejor es la primera opción.
3. Especificar un equipo alternativo como la puerta de enlace, este podría ser el SOC. Este sería el equipo que tenga todas las rutas hacia las diferentes subredes e Internet.

Ejemplo de la configuración de la interfaz de red del SOC llevando a cabo la primera opción de crear una ruta en el equipo (Se puede realizar por terminal o por interfaz gráfica):

```
Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: no
      addresses: [192.168.1.150/24]
      nameservers:
        addresses: [ 8.8.8.8, 8.8.4.4 ]
      routes:
        - to: default
          via: 192.168.1.1
        - to: 192.168.10.0/24
          via: 192.168.1.140
```

Ya tendríamos totalmente comunicadas las máquinas del laboratorio. En esta configuración se ha cambiado el renderer a “networkd”. Tras esta configuración, se deben lanzar los siguientes comandos como root:

```
# systemctl stop NetworkManager
# systemctl disable NetworkManager
# systemctl start systemd-networkd
# systemctl enable systemd-networkd
# netplan apply
```

La última ejecución puede lanzar algunos “Warnings” pero la configuración debe ser correcta y la máquina debería poder comunicarse con el “webserver”. Se puede realizar una prueba de PING.

Nota: La máquina de la red interna WebServer debería tener habilitados los servicios SSH, HTTP y MySQL. Además, tiene instalada la herramienta PHPMysqlAdmin para el acceso a la base de datos.

2.2 Configuración del IDS para registrar el tráfico de red. (SNORT)

El primero paso consiste en la instalación de SNORT:

```
# apt install snort.
```

Habilitamos el servicio con systemctl:

```
# systemctl enable snort.service
```

Tras la instalación, se debe navegar a su directorio de configuración en “/etc/snort”. En este directorio debemos configurar los parámetros básicos de red e interfaz. Para ello, en sistemas debian, editamos el fichero “snort.debian.conf” y configuramos los parámetros HOME_NET e INTERFACE para que sean los valores adecuados de nuestra red. También es posible la configuración directamente del fichero “snort.conf”.

```
DEBIAN_SNORT_STARTUP="boot"
DEBIAN_SNORT_HOME_NET="192.168.10.0/24"
DEBIAN_SNORT_OPTIONS=""
DEBIAN_SNORT_INTERFACE="enp0s8"
DEBIAN_SNORT_SEND_STATS="true"
DEBIAN_SNORT_STATS_RCPT="root"
DEBIAN_SNORT_STATS_THRESHOLD="1"
```

A continuación se definen las reglas de detección de Snort. Estas reglas se reflejan en “/etc/snort/rules”.

En este directorio existen multitud de ficheros con reglas ya predefinidas para la detección de diversos ataques o incidentes. Estas reglas pueden ser ajustadas a los requisitos específicos de cada caso.

Para la creación de reglas propias según nuestras necesidades, editamos el fichero local.rules. Algunos ejemplos de reglas a aplicar:

```
# This file intentionally does not come with signatures. Put your local
# additions here.
alert icmp any any -> $HOME_NET any (msg:"PING exterior detectado."; icode:0; itype:8; sid: 3000001;)
alert icmp HOME_NET any -> any any (msg:"PING interior detectado."; icode:0; itype:8; sid: 3000002;)
alert tcp any any -> 192.168.10.100/24 22 (msg:"Intento de inicio de conexión SSH desde el exterior al servidor web."; flags: S; sid:3000003;)
alert tcp any any -> 192.168.10.100/24 80 (msg:"Intento de entrada a PHPMyadmin"; content: "GET /phpmyadmin"; nocase; sid:3000004;)
alert tcp any any -> 192.168.10.100/24 80 (msg:"Visita a la web del servidor web."; flags: S; sid:3000005;)
```

Para ver en vivo la actividad y registro de estas reglas se debe lanzar el siguiente comando en el IDS:

```
# tail -f /var/log/snort/snort.alert.fast
```

```
lentos@eluserver21: ~$ tail -f /var/log/snort/snort.alert.fast
03/03-10:43:29.916451 [**] [1:3000003:0] Inicio de conexión SSH desde el exterior al servidor web. [**] [Priority: 0] (TCP) 192.168.1.133:51904 -> 192.168.10.100:22
03/03-10:50:40.999102 [**] [1:3000001:0] PING exterior detectado. [**] [Priority: 0] (ICMP) 192.168.1.133 -> 192.168.10.100
03/03-10:50:42.002499 [**] [1:3000001:0] PING exterior detectado. [**] [Priority: 0] (ICMP) 192.168.1.133 -> 192.168.10.100
03/03-10:50:57.146181 [**] [1:3000002:0] PING interior detectado. [**] [Priority: 0] (ICMP) 192.168.10.100 -> 192.168.1.1
03/03-10:50:58.147661 [**] [1:3000002:0] PING interior detectado. [**] [Priority: 0] (ICMP) 192.168.10.100 -> 192.168.1.1
03/03-10:51:05.903398 [**] [1:3000005:0] Visita a la web del servidor web. [**] [Priority: 0] (TCP) 192.168.1.133:51973 -> 192.168.10.100:80
03/03-10:51:42.924837 [**] [1:3000005:0] Visita a la web del servidor web. [**] [Priority: 0] (TCP) 192.168.1.133:51977 -> 192.168.10.100:80
03/03-10:51:42.926942 [**] [1:3000004:0] Intento de entrada a PHPMyadmin [**] [Priority: 0] (TCP) 192.168.1.133:51977 -> 192.168.10.100:80
03/03-10:51:43.475444 [**] [1:3000005:0] Visita a la web del servidor web. [**] [Priority: 0] (TCP) 192.168.1.133:51978 -> 192.168.10.100:80
03/03-10:52:32.943131 [**] [1:3000003:0] Intento de inicio de conexión SSH desde el exterior al servidor web. [**] [Priority: 0] (TCP) 192.168.1.133:51982 -> 192.168.10.100:22
```

Tras los diferentes intentos de comunicación se deben ir visualizando las entradas de este fichero de log en vivo.

3 ELASTIC STACK

La instalación de este conjunto de herramientas (Stack) se realizará con la última versión disponible 8.12. Esta versión hace obligatorio el uso de conexiones seguras entre las herramientas, por lo que la configuración es más compleja que la mostrada en versiones anteriores en las que no se necesitaban el uso de encriptación y certificados.

3.1 Instalación de Elasticsearch.

Para instalar estas herramientas se hará uso del artículo de su página oficial:

<https://www.elastic.co/guide/en/elasticsearch/reference/current/setup.html>

Primero se debe realizar la instalación de Elasticsearch, que se realizará mediante el [paquete de instalación de Debian](#).

Tras los comandos de instalación, se debe mostrar una ventana con información básica de configuración y seguridad. Esta configuración mostrada será: creación de superusuario “elastic” junto con su **clave de acceso (copiar)**, certificados TLS para HTTP, todos los tokens necesarios para Kibana y para incluir nuevos elementos a un clúster que serán válidos por 30 minutos.

```
----- Security autoconfiguration information -----

Authentication and authorization are enabled.
TLS for the transport and HTTP layers is enabled and configured.

The generated password for the elastic built-in superuser is : bG3*U2qR100NnAquIdHf

If this node should join an existing cluster, you can reconfigure this with
'/usr/share/elasticsearch/bin/elasticsearch-reconfigure-node --enrollment-token <token-here>'
after creating an enrollment token on your existing cluster.

You can complete the following actions at any time:

Reset the password of the elastic built-in superuser with
'/usr/share/elasticsearch/bin/elasticsearch-reset-password -u elastic'.

Generate an enrollment token for Kibana instances with
'/usr/share/elasticsearch/bin/elasticsearch-create-enrollment-token -s kibana'.

Generate an enrollment token for Elasticsearch nodes with
'/usr/share/elasticsearch/bin/elasticsearch-create-enrollment-token -s node'.
```

Además, nos indica comandos con los que podremos crear tokens para nuevos nodos de Elasticsearch o para enlazar Kibana.

Tras la instalación, debemos modificar el fichero de configuración para que cree los índices de forma automática. Editamos su fichero de configuración que estaría en “/etc/elasticsearch/elasticsearch.yml”.

Al final de este fichero incluimos la siguiente línea:

```
action.auto_create_index: .monitoring*,.watches,.triggered_watches,.watcher-history*,.ml*,logstash*,filebeat*
```

Esta es una configuración de seguridad para evitar ataques de creación de índices de forma masiva. Se puede cambiar a * y permitiría la creación de todo tipo de índices.

El siguiente paso sería habilitar Elasticsearch como un servicio y lanzarlo. Para ello ejecutamos:

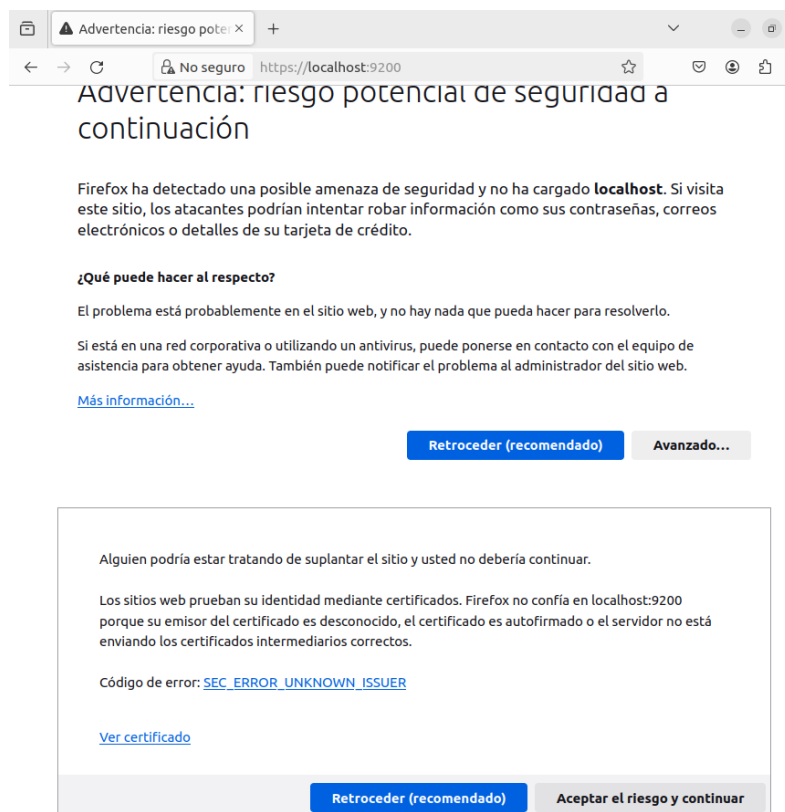
```
# systemctl daemon-reload
```

```
# systemctl enable elasticsearch.service
```

```
# systemctl start elasticsearch.service
```

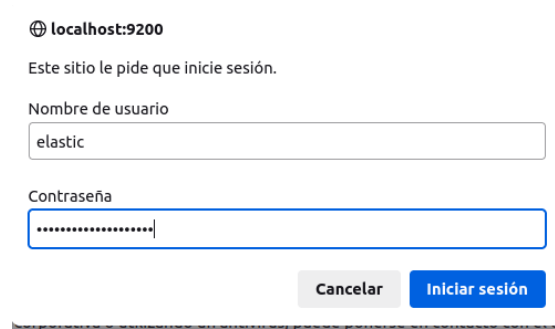
```
# systemctl status elasticsearch.service
```

Ya podríamos acceder al navegador y ver que el servicio funciona:



Nos avisa del riesgo, “Aceptar el riesgo y continuar”.

Nos solicita el usuario y pass (copiado anteriormente):



localhost:9200

Este sitio le pide que inicie sesión.

Nombre de usuario

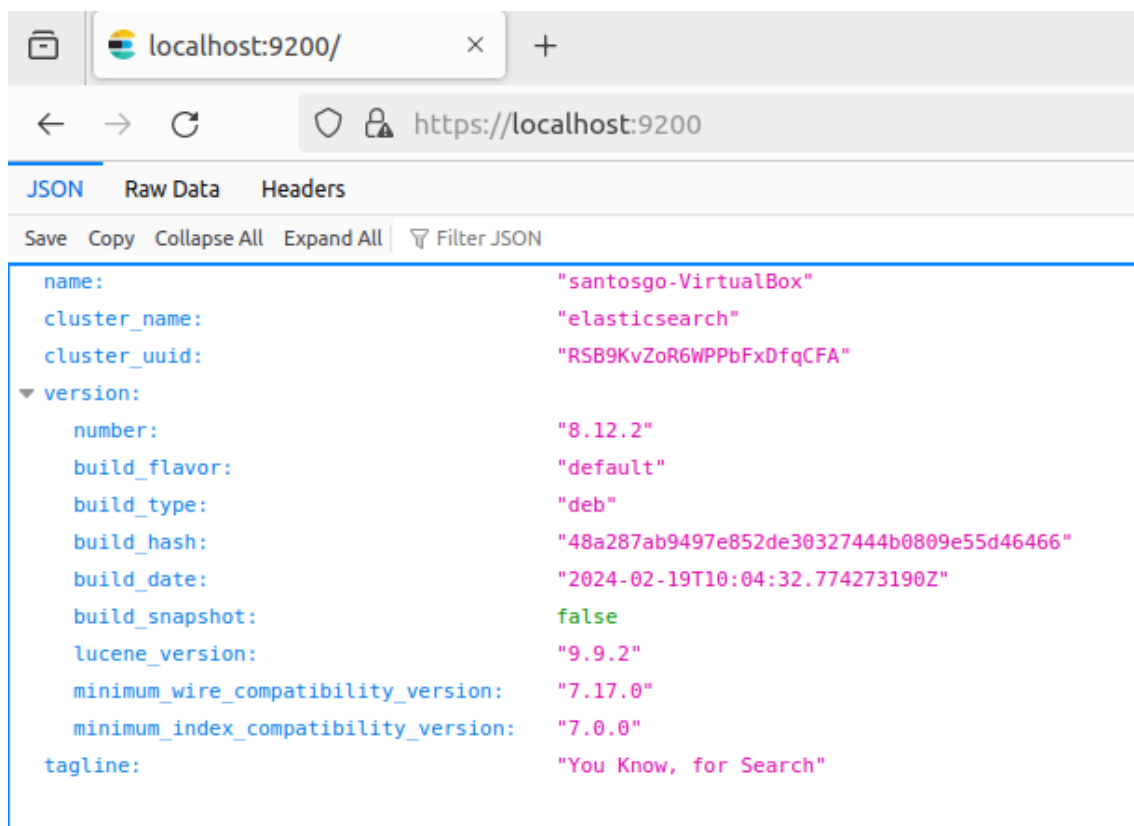
elastic

Contraseña

.....

Cancelar Iniciar sesión

Se concede acceso a la interfaz de Elasticsearch:



localhost:9200/

https://localhost:9200

JSON Raw Data Headers

Save Copy Collapse All Expand All Filter JSON

```
{
  "name": "santosgo-VirtualBox",
  "cluster_name": "elasticsearch",
  "cluster_uuid": "RSB9KvZoR6WPPbFxDfqCFA",
  "version": {
    "number": "8.12.2",
    "build_flavor": "default",
    "build_type": "deb",
    "build_hash": "48a287ab9497e852de30327444b0809e55d46466",
    "build_date": "2024-02-19T10:04:32.774273190Z",
    "build_snapshot": false,
    "lucene_version": "9.9.2",
    "minimum_wire_compatibility_version": "7.17.0",
    "minimum_index_compatibility_version": "7.0.0"
  },
  "tagline": "You Know, for Search"
}
```

También se pueden observar los logs relacionados con este servicio con el siguiente comando:

```
sudo journalctl --unit elasticsearch
```

```
sudo journalctl --unit elasticsearch --since "2016-10-30 18:17:16" # desde un tiempo determinado.
```

También podemos comprobar el correcto funcionamiento del servicio con el comando curl:

```
curl --cacert /etc/elasticsearch/certs/http_ca.crt -u elastic:ELASTIC_PASSWORD https://localhost:9200
```

```

root@santosgo-VirtualBox:/etc/elasticsearch# curl --cacert /e
{
  "name" : "santosgo-VirtualBox",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "RSB9KvZoR6WPPbFxDfqCFA",
  "version" : {
    "number" : "8.12.2",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "48a287ab9497e852de30327444b0809e55d46466",
    "build_date" : "2024-02-19T10:04:32.774273190Z",
    "build_snapshot" : false,
    "lucene_version" : "9.9.2",
    "minimum_wire_compatibility_version" : "7.17.0",
    "minimum_index_compatibility_version" : "7.0.0"
  },
  "tagline" : "You Know, for Search"
}

```

El directorio de configuración de Elasticsearch pertenece a root:elasticsearch.

Una vez que se ha instalado y configurado los aspectos básicos de esta herramienta es el momento de profundizar en la configuración.

Las principales rutas de esta herramienta son:

Datos: /var/data/elasticsearch

Logs: /var/log/elasticsearch

Configuración: /etc/elasticsearch/elasticsearch.yml

Por defecto, **Elasticsearch es solamente accesible desde el propio equipo en el que se instala**, pero si el objetivo es exponer el sistema a la red, se tendrá que configurar el parámetro “network.host” de su fichero de configuración indicando su dirección IP de escucha. Esta configuración solo es necesaria si los datos enviados por Filebeat serán procesados directamente por esta herramienta, ya que en caso de que Logstash sea quien recoja los datos pues este enviará los datos a Elasticsearch desde la propia máquina, por lo que no será necesaria su configuración.

Para casos de empresas con una vital importancia de la seguridad de los datos, esta herramienta provee del uso de “keystore” para el acceso a sus recursos.

Además, por temas de seguridad, debemos modificar el fichero de configuración para que se restrinja la creación de los índices de forma automática. Editamos su fichero de configuración e incluimos la siguiente línea:

```

action.auto_create_index: .monitoring*,.watches,.triggered_watches,.watcher-history*,.ml*,
filebeat*, logstash*

```

Esta es una configuración de seguridad para evitar ataques de creación de índices de forma masiva. Se puede cambiar a * y permitiría la creación de todo tipo de índices. A la hora de crear los índices en los agentes se debe tener en cuenta que estos índices deben cuadrar con el algún patrón de los permitidos. En caso contrario, dará un error en la creación del índice.

3.2 Instalación y configuración de Kibana.

El primer paso será la instalación de Kibana mediante su [paquete Debian](#) (Debian package).

Tras su instalación, tendremos que generar un token de elasticsearch para su enlace con Kibana.

```
$ usr/share/elasticsearch/bin/elasticsearch-create-enrollment-token -s kibana
```

```
root@santoso-VirtualBox: /etc/elasticsearch# /usr/share/elasticsearch/bin/elasticsearch-create-enrollment-token -s kibana
eyJ2Z2xiOjI0IjAlJjEYLjllLCh2HI0LsIMTKyLjE2OC4xLjE3MDs5MjAwI0sInZnc1I6IjllZDVlODBhYmF0TU82Tg1ODUxMzYwMzBhYWE6ZWFkNDk1ZDA1ZmMyY2F1NjRhZWY2ZjFUMWNNKjQwMmULLC3rZXk1O1IybnJkQ280bn34RfpxbntCVFNZ0Dp2Wip3eGZP
aFFqTy10CXRMTUowVnF3In0=
```

Tras generar este token se puede lanzar el comando de enlace de kibana con elasticsearch:

```
$ /usr/share/kibana/bin/kibana-setup --enrollment-token "token_generado".
```

Al finalizar, se debe mostrar un mensaje indicando que la configuración se ha establecido correctamente. Además, se puede visualizar esta configuración en el fichero `/etc/kibana/kibana.yml`:

```
## >>>>> BACKUP END: Kibana interactive setup (2024-03-05T14:28:04.530Z)
# This section was automatically generated during setup.
server.port: 5601
server.host: localhost
elasticsearch.hosts: ['https://192.168.1.150:9200']
elasticsearch.serviceAccountToken: AAEAMWvsYXN0aWVva2llVW5hL2Vucn9sbC1ucn9jZXNzLXRva2VulTE3MDk2NDg0MDAwMDk6a2h3Z29VNEhtbHkSWTQ0VTN1bW5oZW
logging.appenders.file.type: file
logging.appenders.file.filename: /var/log/kibana/kibana.log
logging.appenders.file.layout.type: json
logging.root.appenders: [default, file]
pid.file: /run/kibana/kibana.pid
elasticsearch.ssl.certificateAuthorities: [/var/lib/kibana/ca.1789948404524.crt]
xpack.fleet.outputs: [{id: fleet-default-output, name: default, is_default: true, type: elasticsearch, hosts: ['https://192.168.1.150:9200'], ca_trusted_fingerprint: 9ed5b80ab
a0954e838513f036f4a4ead415d05ff01c2cae04aef0f1b2ca041ze}]
```

A continuación, se habilitará Kibana como un servicio y se iniciará:

```
# systemctl daemon-reload
```

```
# systemctl enable kibana.service
```

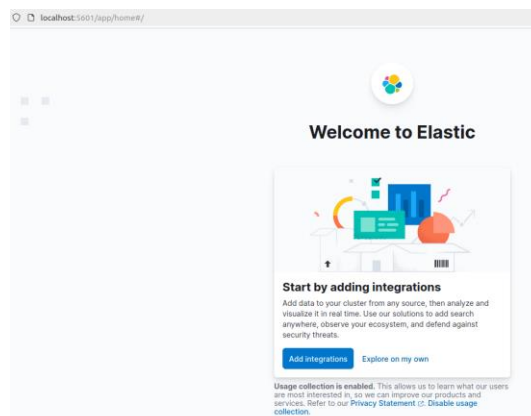
```
# systemctl start kibana.service
```

```
# systemctl status kibana.service
```

Tras el inicio de Kibana hay que tener paciencia y esperar unos 10 minutos para que levante correctamente todo el servicio por primera vez.

Se puede observar el servicio en <http://localhost:5601>

Nos aparecerá una ventana como la siguiente en la que se nos solicitará el usuario y la clave del usuario de Elasticsearch configurados anteriormente. Una vez autenticado, ya se puede acceder a su menú:



3.3 Instalación de FileBeats en los diferentes agentes IDS y principales servidores.

Para la instalación de FileBeats podemos acceder a la web de ELK y buscar la zona de FileBeats.

Aquí os dejo el enlace directo a la parte de instalación:

<https://www.elastic.co/guide/en/beats/filebeat/current/filebeat-installation-configuration.html>

Se debe elegir el tipo de instalación “Self-managed” y posteriormente elegir el tipo de sistema operativo. En este caso elegiremos para sistemas DEB. Se deben ejecutar los siguientes comandos:

```
curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-8.12.2-amd64.deb
sudo dpkg -i filebeat-8.12.2-amd64.deb
```

Tras la instalación en el sistema siguiendo las indicaciones de la web, debemos proceder a:

1. Habilitación del servicio: `systemctl enable filebeat.service`.
2. Arranque del servicio: `systemctl start filebeat.service`.
3. Comprobación del servicio: `systemctl status filebeat.service`.
4. Observar los log de filebeat: `journalctl -u filebeat.service`.

El fichero de configuración de FileBeat está en “/etc/filebeat/filebeat.yml”.

FileBeat solamente permite la salida a un único destino, por lo que en este caso configuraremos una salida por pantalla para la realización de unas primeras pruebas. Tras comprobar el correcto funcionamiento de este servicio, la salida será redirigida a Logstash de la máquina SOC-SIEM.

3.3.1 Configuración básica para salida por pantalla:

Editamos el fichero de filebeat.yml con los siguientes datos:

- La entrada debe ser el fichero de log de Snort y esta debe ser “habilitada” ya que por defecto viene deshabilitada:

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

- La salida la configuramos inicialmente a consola:

```
output.console:
  pretty: true
# =====
```

- Hay que comentar la salida a elasticsearch, ya que filebeat solamente permite una salida activa:

```
# ----- Elasticsearch Output -----
#output.elasticsearch:
#  # Array of hosts to connect to.
#  #hosts: ["192.168.1.150:9200"]

#  # Performance preset - one of "balanced", "throughput", "scale",
#  # "latency", or "custom".
#  #preset: balanced

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

#  # Authentication credentials - either API key or username/password.
#  #api_key: "AAEAAWVsYXN0aWVva2liYW5hL2Vucm9sbC1wcm9jZjZlXzNzLXRva2VuLTE3MDk2NDg0MDEwMDk6azhqZ29VNEhTbHk5WTQ0VTNiW5oZWw"
#  #username: "elastic"
#  #password: "bG3*U2qR100NnAquIdHf"
```

Comprobamos que el fichero de configuración no contiene errores con el siguiente comando:

```
root@luserver22:/etc/filebeat# filebeat test config -c filebeat.yml
Config OK
```

Lanzamos la ejecución de filebeat para comprobar que funciona correctamente y que muestra los logs de Snort:

```
root@luserver22:/etc/filebeat# filebeat -c filebeat.yml
```

Si se comienzan a realizar operaciones que propician el registro de logs en Snort al poco tiempo aparecerán en la siguiente consola en formato JSON.

```
root@luserver22:/etc/filebeat# filebeat -c filebeat.yml
{"@timestamp": "2024-03-17T11:24:03.983Z",
"@metadata": {
  "beat": "filebeat",
  "type": "_doc",
  "version": "8.12.2"
},
"log": {
  "offset": 257800,
  "file": {
    "inode": "132564",
    "path": "/var/log/snort/snort.alert.fast",
    "device_id": "64768"
  }
},
"message": "03/17-11:23:55.848661  [**] [1:300002:0] PING interior detectado. [**] [Priority: 0] {ICMP} 192.168.10.100 -> 192.168.1.1",
"input": {
  "type": "filestream"
},
"ecs": {
  "version": "8.0.0"
},
"host": {
  "hostname": "luserver22",
  "architecture": "x86_64",
  "os": {
    "family": "debian",
    "name": "Ubuntu",
    "kernel": "5.15.0-97-generic",
    "codename": "jammy",
    "type": "linux",
    "platform": "ubuntu",
    "version": "22.04.4 LTS (Jammy Jellyfish)"
  },
  "id": "793ca49c79dc4fbf83b36cef40694872",
  "containerized": false,
  "ip": [
    "192.168.1.140",
    "fe80::a00:27ff:fe19:8726",
    "192.168.10.1",
    "fe80::a00:27ff:fe41:f95"
  ],
  "mac": [
    "08-00-27-19-87-26",
    "08-00-27-41-0F-95"
  ]
},
}
```

3.3.2 Habilitar un módulo en Filebeat:

Filebeat puede trabajar con diferentes módulos preinstalados para observar e ingestar sus logs. Por defecto estos módulos están desactivados. La lista de módulos se puede observar con el comando:

```
# filebeat modules list
```

Nos mostraría toda la lista, primero los habilitados y posteriormente los deshabilitados. Como inicialmente no existen módulos habilitados, pues solamente muestra la lista con todos los módulos.

Para habilitar un módulo se puede usar el comando:

```
# filebeat modules enable nombre_modulo
```

Posteriormente se puede acceder al fichero de configuración del módulo que se encontraría en `/etc/filebeat/modules.d/`. El fichero del módulo habilitado se debe editar ya que, aunque se habilita el módulo, este tiene deshabilitadas sus listas de logs. En el fichero se indica a `"true"` los diferentes logs que queramos habilitar y podríamos indicar la lista de ficheros de log personalizada si queremos. Si la lista se deja vacía, Filebeat determinará los ficheros de log a procesar para el módulo de forma automática.

```
# Module: apache
# Docs: https://www.elastic.co/guide/en/beats/filebeat/8.12/filebeat-module-apache.html

- module: apache
  # Access logs
  access:
    enabled: true

  # Set custom paths for the log files. If left empty,
  # Filebeat will choose the paths depending on your OS.
  #var.paths:

  # Error logs
  error:
    enabled: true

  # Set custom paths for the log files. If left empty,
  # Filebeat will choose the paths depending on your OS.
  #var.paths:
```

Con esta configuración Filebeat ya estaría preparado para enviar los datos de log del servicio Apache, tanto de `"acceso"` como de `"error"`.

Podemos ver cómo efectivamente se registran los eventos de Apache:

```
{
  "@timestamp": "2024-03-17T11:45:10.903Z",
  "@version": 1,
  "beat": {
    "name": "lucifer22",
    "type": "filebeat",
    "version": "8.12.2",
    "pipeline": "filebeat-8.12.2-apache-access-pipeline"
  },
  "ecp": {
    "version": "1.12.0"
  },
  "agent": {
    "id": "d1e5073b-e344-4399-971b-5733c52f1ed",
    "name": "lucifer22",
    "type": "filebeat",
    "version": "8.12.2",
    "ephemeral_id": "90d55bc8-757b-4288-83d9-6a7ed4be4843"
  },
  "message": "192.168.1.134 - - [17/Mar/2024:11:45:10 -0000] \"GET /icons/ubuntu-logo.png HTTP/1.1\" 200 3607 \"http://192.168.10.1/\"",
  "service": {
    "type": "apache"
  },
  "input": {
    "type": "log"
  },
  "host": {
    "ip": [
      "192.168.1.140",
      "fe80::a00:27ff:fe19:8726",
      "192.168.10.1",
      "fe80::a00:27ff:fe01:f95"
    ],
    "mac": [
      "08-00-27-19-87-26",
      "08-00-27-41-8f-25"
    ],
    "hostname": "lucifer22",
    "name": "lucifer22",
    "architecture": "x86_64",
    "os": {
      "name": "Ubuntu",
      "kernel": "5.15.0-97-generic",
      "codename": "jammy",
      "type": "linux",
      "platform": "ubuntu",
      "version": "22.04.4 LTS (Jammy Jellyfish)",
      "family": "debian"
    },
    "id": "783ca49c79dc4f9f8335ccf48094872",
    "containerized": false
  },
  "log": {
    "offset": 1156,
    "file": {
      "path": "/var/log/apache2/access.log"
    }
  },
  "event": {
    "module": "apache",
    "dataset": "apache_access"
  },
  "fileset": {
    "name": "access"
  }
}
```

3.3.3 Envío de datos a Logstash:

Para el envío de datos desde FileBeat debemos cambiar la salida, comentando la salida que tuviéramos configurada anteriormente y descomentando la salida a logstash indicando la dirección IP de la máquina SOC. el puerto será el que viene por defecto, 5044.

Ejemplo:

```
# ----- Logstash Output -----  
output.logstash:  
  # The Logstash hosts  
  hosts: ["192.168.1.150:5044"]
```

A continuación, podemos lanzar Filebeat con el comando usado anteriormente o lanzando el servicio (preferiblemente el servicio).

Filebeat -c filebeat.yml

Service filebeat start

El siguiente paso que tenemos que conseguir es la instalación y configuración inicial de Logstash. Esta herramienta la instalaremos en el propio SOC gracias al panel de Kibana, ya instalado y configurado anteriormente.

3.4 Instalación de Logstash.

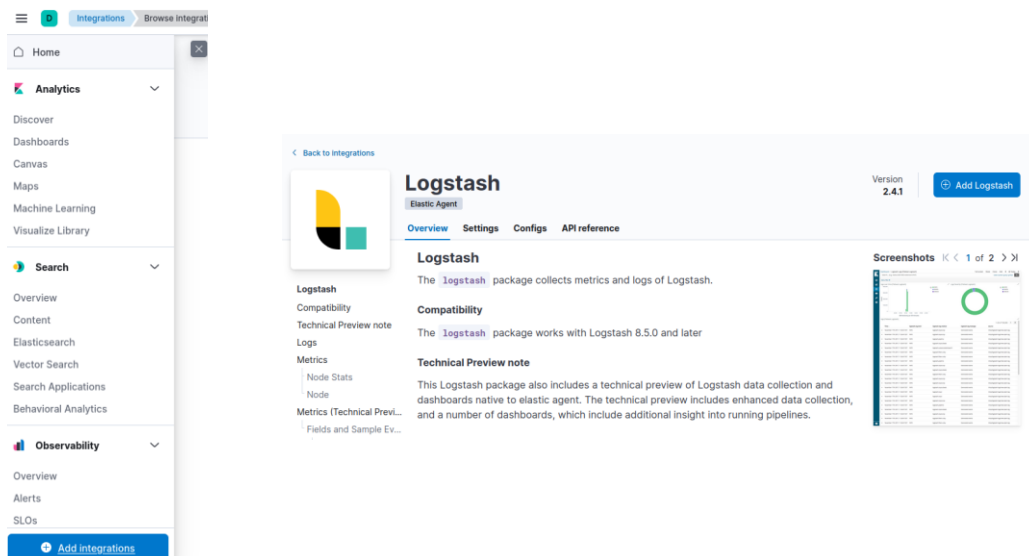
Tras la visualización del panel de gestión web de Kibana, se ha descubierto que se pueden instalar muchas integraciones desde esta propia app.

**Nota: Logstash necesita de la instalación del jdk java11. Puede que si no está instalado en la máquina este software no funcione correctamente. Hay que comprobar si está instalado.*

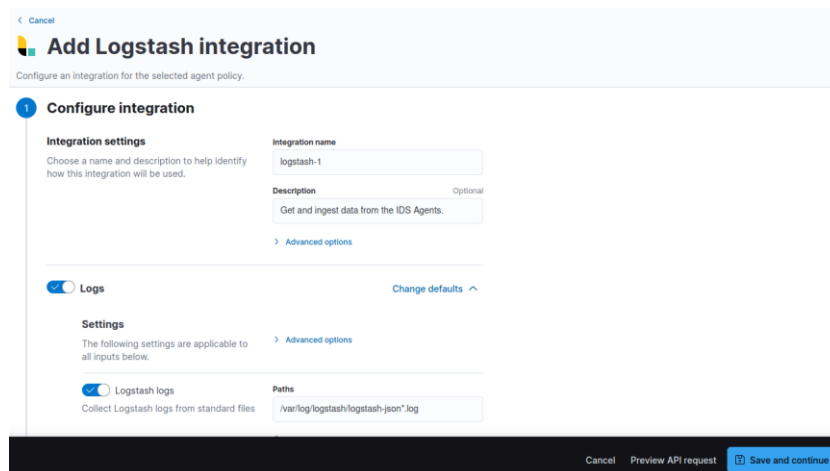
Se puede instalar con el siguiente comando:

apt install openjdk-11-jre-headless

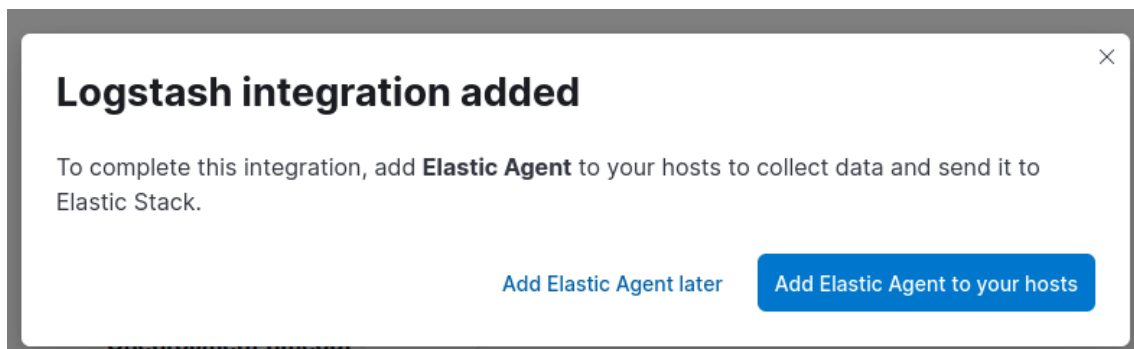
A continuación, se va a llevar a cabo la instalación de Logstash desde la web, añadiendo una nueva integración (Add integrations). Buscamos el término “Logstash” y lo instalamos.



Escribimos un nombre y una descripción y guardamos:



Por último, nos pide confirmación:



A continuación, aparecen unos pasos para incluir unos datos de configuración en el fichero de configuración de “logstash” (/etc/logstash/logstash.yml). Estos pasos de configuración no son necesarios realizarlos, puesto que realizaremos una configuración manual para el acceso seguro a la escritura de logs.

El segundo paso sería instalar el agente de Elastic (Este paso no es necesario realizarlo):

2 Install Elastic Agent on your host

Select the appropriate platform and run commands to install, enroll, and start Elastic Agent. Reuse commands to set up agents on more than one host. For aarch64, see our [downloads page](#). This guidance is for AMD but you can adapt it to your device architecture. For additional guidance, see our [installation docs](#).

Linux Tar Mac Windows RPM DEB

```
curl -L -O https://artifacts.elastic.co/downloads/beats/elastic-agent/elastic-agent-8.12.2
tar xzvf elastic-agent-8.12.2-linux-x86_64.tar.gz
cd elastic-agent-8.12.2-linux-x86_64
sudo ./elastic-agent install
```

Si no se realiza la instalación del agente (opcional) o no se ha instalado Logstash por algún motivo, realizamos la instalación de Logstash a nivel del sistema con el comando:

```
# apt install logstash
```


Para comprobar que logstash se ejecuta correctamente y funciona, se puede hacer uso del siguiente comando:

```
/usr/share/logstash/bin/logstash -e 'input{ stdin{} } output { stdout{} }'
```

Entonces, cuando termina la carga, podemos escribir lo que queramos y debe mostrarse la cadena ya en el formato propio JSON.

```
root@santoso-VirtualBox:/etc/logstash/elastic-agent-8.12.2-linux-x86_64# /usr/share/logstash/bin/logstash -e 'input{ stdin{} } output { stdout{} }'
Using bundled JRE: /usr/share/logstash/jdk
/usr/share/logstash/vendor/bundle/ruby/3.1.0/gems/concurrent-ruby-1.1.9/lib/concurrent-ruby/concurrent/executor/java_thread_pool_executor.rb:13: warning: method redefined; discarding old to_int
/usr/share/logstash/vendor/bundle/ruby/3.1.0/gems/concurrent-ruby-1.1.9/lib/concurrent-ruby/concurrent/executor/java_thread_pool_executor.rb:13: warning: method redefined; discarding old to_f
WARNING: Could not find logstash.yml which is typically located in $LS_HOME/config or /etc/logstash. You can specify the path using --path.settings. Continuing using the defaults
Could not find log4j2 configuration at path /usr/share/logstash/config/log4j2.properties. Using default config which logs errors to the console
[WARN ] 2024-03-05 18:06:56.274 [main] runner - NOTICE: Running Logstash as superuser is not recommended and won't be allowed in the future. Set 'allow_superuser' to 'false' to avoid startup errors in fut
ure releases.
[INFO ] 2024-03-05 18:06:56.351 [main] runner - Starting Logstash ("logstash.version"=>"8.12.2", "jruby.version"=>"jruby 9.4.5.0 (3.1.4) 2023-11-02 1abae2700f OpenJDK 64-Bit Server VM 17.0.10+7 on 17.0.10
+7 +indy +jit [x86_64-linux]")
[INFO ] 2024-03-05 18:06:56.368 [main] runner - JVM bootstrap flags: [-XX:HeapDumpOnOutOfMemoryError, -Dlogstash.jackson.stream.read.constraints.max-number-length=10000, --add-opens=java.base/java.nio.ch
annels=ALL-UNNAMED, --add-exports=jdk.compiler/com.sun.tools.javac.tree=ALL-UNNAMED, -Djruby.exexp.intererruptible=true, --add-opens=java.base/java.security=ALL-UNNAMED, --add-exports=jdk.compiler/com.sun
tools.javac.util=ALL-UNNAMED, --add-exports=jdk.compiler/com.sun.tools.javac.parser=ALL-UNNAMED, --add-opens=java.management/sun.management=ALL-UNNAMED, --add-exports=jdk.compiler/com.sun.tools.javac.file
=ALL-UNNAMED, -Dio.nettyallocator.maxOrder=11, -Dlog4j2.isThreadContextMapInherited=true, -Xmxsig, -Dlogstash.jackson.stream.read.constraints.max-string-length=200000000, -Djdk.io.File.enableADS=true, -
Dfile.encoding=UTF-8, --add-opens=java.base/java.lang=ALL-UNNAMED, --add-exports=jdk.compiler/com.sun.tools.javac.api=ALL-UNNAMED, -Djruby.compile.invokedynamic=true, -Xmxsig, -Djava.security.egd=file:/dev/ur
andom, -Djava.aot.headless=true, --add-opens=java.base/sun.nio.ch=ALL-UNNAMED]
[INFO ] 2024-03-05 18:06:56.368 [main] runner - Jackson default value override 'logstash.jackson.stream.read.constraints.max-string-length' configured to '200000000'
[INFO ] 2024-03-05 18:06:56.428 [main] runner - Jackson default value override 'logstash.jackson.stream.read.constraints.max-number-length' configured to '10000'
[WARN ] 2024-03-05 18:06:58.238 [Logstash:Runner] multilocal - Ignoring the 'pipelines.yml' file because modules or command line options are specified
[INFO ] 2024-03-05 18:07:02.868 [agent - Successfully started Logstash API endpoint (:port=>9600, ssl_enabled=>false)]
[INFO ] 2024-03-05 18:07:04.085 [Converge PipelineAction::Create=main] Reflections - Reflections took 941 ns to scan 1 urls, producing 132 kops and 468 values
/usr/share/logstash/vendor/bundle/ruby/3.1.0/gems/maazing-print-1.5.0/lib/maazing-print/formatter.rb:37: warning: previous definition of cast was here
[INFO ] 2024-03-05 18:07:06.912 [Converge PipelineAction::Create=main] jvavapipeline - Pipeline 'main' is configured with 'pipeline.ecs_compatibility: v8' setting. All plugins in this pipeline will default
to 'ecs_compatibility => v8' unless explicitly configured otherwise.
[INFO ] 2024-03-05 18:07:07.134 [[main] pipeline-manager] jvavapipeline - Starting pipeline (:pipeline_id=>"main", "pipeline.workers"=>1, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>50, "pipeline.n
ax_inflight"=>125, "pipeline.sources"=>["config string"], :threads=>#<Thread:0x49ce3782 /usr/share/logstash/logstash-core/lib/logstash/java_pipeline.rb:134 run>")
[INFO ] 2024-03-05 18:07:11.557 [[main] pipeline-manager] jvavapipeline - Pipeline Java execution initialization time ("seconds"=>4.41)
[INFO ] 2024-03-05 18:07:11.779 [[main] pipeline-manager] jvavapipeline - Pipeline started ("pipeline.id"=>"main")
The stdin plugin is now waiting for input:
[INFO ] 2024-03-05 18:07:11.861 [agent thread agent - Pipelines running (:count=>1, :running_pipelines=>[main], :non_running_pipelines=>[])]
hola
{
  "host" => {
    "hostname" => "santoso-VirtualBox"
  },
  "@version" => "1",
  "@timestamp" => 2024-03-05T17:07:30.481436359Z,
  "event" => {
    "original" => "hola"
  },
  "message" => "hola"
}
```

Tras esta comprobación, se puede observar el servicio (service). Además, se puede habilitar el servicio como se ha hecho con las herramientas anteriores.

Hasta ahora, tenemos configurado nuestro agente IDS con el servicio Filebeat activo para el envío de datos hacia Logstash.

3.4.1 Configuración de un pipeline:

El siguiente paso sería “Configurar un Pipeline en Logstash” (camino o tubo de ingesta). Logstash utiliza pipelines para procesar los datos entrantes. Debes crear un archivo de configuración de pipeline que especifique cómo deseas que Logstash procese los logs. Por lo general, estos archivos de configuración tienen la extensión .conf y se almacenan en el directorio de configuración de Logstash (/etc/logstash/conf.d/). Todos los pipelines de este directorio se habilitarían con el lanzamiento de Logstash.

Por ejemplo, crea un archivo llamado first_pipeline.conf en el directorio de configuración de Logstash. Puedes usar un editor de texto para crear y editar este archivo.

Configurar el Pipeline para Recibir Logs de Filebeat:

Dentro del archivo first_pipeline.conf, configura un input para recibir los logs de Filebeat. Puedes configurar Logstash para escuchar en el puerto especificado por Filebeat para recibir los logs.

Por ejemplo, para configurar Logstash para escuchar en el puerto 5044:

```
input {
  beats {
    port => 5044
  }
}
```

Definir Filtros y Salidas:

Después de recibir los logs, es posible que desees aplicar filtros para analizar, enriquecer o modificar los datos según tus necesidades. Los filtros en Logstash se utilizan para este propósito. Por ejemplo, puedes agregar un filtro para parsear los logs y estructurarlos de manera adecuada. Esta parte es opcional.

```
filter {  
  # Aquí puedes agregar los filtros necesarios  
}
```

Luego, define una salida donde desees enviar los logs procesados. Puedes enviarlos a Elasticsearch, a un archivo, a un servicio de almacenamiento en la nube, entre otros destinos.

Por ejemplo, para enviar los logs procesados a consola:

```
output {  
  stdout { codec => rubydebug }  
}
```

El fichero quedaría así (se encuentra fuera del directorio conf.d ya que es de prueba):

```
santosgo@santosgo-VirtualBox:~$ sudo cat /etc/logstash/first_pipeline.conf  
input {  
  beats {  
    port => "5044"  
  }  
}  
# The filter part of this file is commented out to indicate that it is  
# optional.  
# filter {  
#  
#}  
output {  
  stdout { codec => rubydebug }  
}
```

Iniciar Logstash:

Una vez que hayas configurado el archivo first_pipeline.conf, podemos comprobar su correcta configuración con el siguiente comando, dando como resultado OK:

```
# /usr/share/logstash/bin/logstash -f first_pipeline.conf --config.test_and_exit
```

```
rootsantosgo-VirtualBox:/etc/logstash# /usr/share/logstash/bin/logstash -f first_pipeline.conf --config.test_and_exit  
Using bundled JDK: /usr/share/logstash/jdk  
[WARN] 2024-03-17 14:02:10.463 [main] runner - NOTICE: Running Logstash as Superuser is not recommended and won't be allowed in the future. Set 'allow_superuser' to 'false' to avoid startup errors releases.  
[INFO] 2024-03-17 14:02:10.463 [main] runner - Starting Logstash ["logstash.version"=>"8.12.2", "jruby.version"=>"jruby 9.4.5.0 (3.1.4) 2023-11-02 1abae2708f OpenJDK 64-Bit Server VM 17.0.10+7  
+7 +indy +jit [x86_64-linux]]"  
[INFO] 2024-03-17 14:02:10.467 [main] runner - JVM bootstrap flags: [-XX:+HeapDumpOnOutOfMemoryError, -Dlogstash.jackson.stream.read-constraints.max-number-length=10000, --add-opens=java.base/j  
annels=ALL-UNNAMED, --add-exports=jdk.compiler/com.sun.tools.javac.tree=ALL-UNNAMED, -Djruby.regex.interruptible=true, --add-opens=java.base/java.security=ALL-UNNAMED, --add-exports=jdk.compiler  
tools.javac.util=ALL-UNNAMED, --add-exports=jdk.compiler/com.sun.tools.javac.parser=ALL-UNNAMED, --add-opens=java.management/sun.management=ALL-UNNAMED, --add-exports=jdk.compiler/com.sun.tools  
ALL-UNNAMED, -Dio.netty.allocation.maxOrder=11, -Dlog4j2.isThreadContextMapInheritable=true, -Xms1g, -Dlogstash.jackson.stream.read-constraints.max-string-length=20000000, -Djdk.io.file.enableA  
Offle.encoding=UTF-8, --add-opens=java.base/java.io=ALL-UNNAMED, --add-exports=jdk.compiler/com.sun.tools.javac.api=ALL-UNNAMED, -Djruby.compile.invokedynamic=true, -Xmx1g, -Djava.security.egd=f  
random, -Djava.awt.headless=true, --add-opens=java.base/sun.nio.ch=ALL-UNNAMED]  
[INFO] 2024-03-17 14:02:10.469 [main] runner - Jackson default value override 'logstash.jackson.stream.read-constraints.max-string-length' configured to '200000000'  
[INFO] 2024-03-17 14:02:10.470 [main] runner - Jackson default value override 'logstash.jackson.stream.read-constraints.max-number-length' configured to '10000'  
[WARN] 2024-03-17 14:02:10.986 [Logstash:Runner] multilocal - Ignoring the 'pipelines.yml' file because modules or command line options are specified  
[INFO] 2024-03-17 14:02:10.993 [Logstash:Runner] configpathloader - No config files found in path {:path=>"/usr/share/logstash/first_pipeline.conf"}  
[ERROR] 2024-03-17 14:02:10.995 [Logstash:Runner] sourceloader - No configuration found in the configured sources.  
Configuration OK  
[INFO] 2024-03-17 14:02:10.996 [Logstash:Runner] runner - Using config.test_and_exit mode. Config Validation Result: OK. Exiting Logstash
```

Si todo ha ido correcto podemos lanzar su ejecución con el comando:

```
# /usr/share/logstash/bin/logstash -f /etc/logstash/first_pipeline.conf --config.reload.automatic
```

```

root@antago-virtualbox:/etc/logstash# /usr/share/logstash/bin/logstash -f /etc/logstash/first_pipeline.conf --config.reload.automatic
Using bundled JDK: /usr/share/logstash/jdk
/usr/share/logstash/vendor/bundle/ruby/3.1.0/gems/concurrent-ruby-1.1.9/lib/concurrent-ruby/concurrent/executor/java_thread_pool_executor.rb:13:
/usr/share/logstash/vendor/bundle/ruby/3.1.0/gems/concurrent-ruby-1.1.9/lib/concurrent-ruby/concurrent/executor/java_thread_pool_executor.rb:13
WARNING: Could not find log4j2 configuration at path /usr/share/logstash/config/log4j2.properties. Using default config which logs errors to the console.
[WARN ] 2024-03-17 14:08:34.129 [main] runner - NOTICE: Running Logstash as superuser is not recommended and won't be allowed in the future. Set
ure releases.
[INFO ] 2024-03-17 14:08:34.273 [main] runner - Starting Logstash {"logstash.version"=>"8.12.2", "jruby.version"=>"jruby 9.4.5.0 (3.1.4) 2023-11
7-rindy with [x86_64-linux]"}
[INFO ] 2024-03-17 14:08:34.277 [main] runner - JVM bootstrap flags: [-XX:+HeapDumpOnOutOfMemoryError, -Dlogstash.jackson.stream.read-constraint
annels=ALL-UNNAMED, --add-exports=jdk.compiler/com.sun.tools.javac.tree=ALL-UNNAMED, -Djruby.regexp.interruptible=true, --add-opens=java.base/ja
tools.javac.util=ALL-UNNAMED, --add-exports=jdk.compiler/com.sun.tools.javac.parser=ALL-UNNAMED, --add-opens=java.management/sun.management=ALL
-ALL-UNNAMED, -Dio.netty.allocator.maxOrder=11, -Dlog4j2.isThreadContextMapInheritable=true, -Xnsig, -Dlogstash.jackson.stream.read-constraints.
Dfile.encoding=UTF-8, --add-opens=java.base/java.io=ALL-UNNAMED, --add-exports=jdk.compiler/com.sun.tools.javac.api=ALL-UNNAMED, -Djruby.compile
random, -Djava.awt.headless=true, --add-opens=java.base/sun.nio.ch=ALL-UNNAMED]
[INFO ] 2024-03-17 14:08:34.279 [main] runner - Jackson default value override 'logstash.jackson.stream.read-constraints.max-string-length' conf
[INFO ] 2024-03-17 14:08:34.280 [main] runner - Jackson default value override 'logstash.jackson.stream.read-constraints.max-number-length' conf
[WARN ] 2024-03-17 14:08:35.009 [Logstash::Runner] multilocal - Ignoring the 'pipelines.yml' file because modules or command line options are sp
[INFO ] 2024-03-17 14:08:36.418 [Api Webserver] agent - Successfully started Logstash API endpoint {:port=>9600, :ssl_enabled=>false}
[INFO ] 2024-03-17 14:08:36.547 [Converge PipelineAction::Create=main] Reflections - Reflections took 259 ms to scan 1 urls, producing 132 keys
/usr/share/logstash/vendor/bundle/ruby/3.1.0/gems/amazing_print-1.5.0/lib/amazing_print/formatter.rb:37: warning: previous definition of cast w
[INFO ] 2024-03-17 14:08:37.872 [Converge PipelineAction::Create=main] Javapipline - Pipeline 'main' is configured with 'pipeline.ecs_compatib
t to 'ecs_compatibility' => v8' unless explicitly configured otherwise.
[INFO ] 2024-03-17 14:08:37.982 [main] pipeline-manager] Javapipline - Starting pipeline {:pipeline_id=>"main", "pipeline.workers"=>4, "pipeli
ex_inflight"=>590, "pipeline.sources"=>["/etc/logstash/first_pipeline.conf"]}, :thread=>#<Thread:0x4e0ff86 /usr/share/logstash/logstash-core/li
[INFO ] 2024-03-17 14:08:38.842 [main] pipeline-manager] Javapipline - Pipeline Java execution initialization time {"seconds"=>0.88}
[INFO ] 2024-03-17 14:08:38.851 [main] pipeline-manager] beats - Starting input listener {:address=>"0.0.0.0:5044"}
[INFO ] 2024-03-17 14:08:38.893 [main] pipeline-manager] Javapipline - Pipeline started {:pipeline_id=>"main"}
[INFO ] 2024-03-17 14:08:38.928 [Agent thread] agent - Pipelines running {:count=>1, :running_pipelines=>{:main}, :non_running_pipelines=>[]}
[INFO ] 2024-03-17 14:08:39.067 [main] beats] Server - Starting server on port: 5044

```

El servicio se queda a la espera de los registros de Filebeat, escuchando por el puerto 5044. Si se realizan pruebas de comunicación que detecte el IDS, los logs de SNORT se deben mostrar en este servicio.

Se puede comprobar la información que ingesta:

```

{
  "tags" => [
    [0] "beats_input_codec_plain_applied"
  ],
  "input" => {
    "type" => "filestream"
  },
  "@version" => "1",
  "ecs" => {
    "version" => "8.0.0"
  },
  "agent" => {
    "id" => "d1e5073b-e3f4-4399-971b-5733cb52f1ed",
    "ephemeral_id" => "5d8a500f-f60b-40fe-a9c4-b165b8911761",
    "type" => "filebeat",
    "name" => "luserver22",
    "version" => "8.12.2"
  },
  "@timestamp" => 2024-03-17T13:11:07.952Z,
  "message" => "03/17-13:11:00.412733  [**] [1:3000004:0] Intento de entrada a PHPMyadmin [**]
  "event" => {
    "original" => "03/17-13:11:00.412733  [**] [1:3000004:0] Intento de entrada a PHPMyadmin [**]
  },
  "host" => {
    "mac" => [
      [0] "08-00-27-19-87-26",
      [1] "08-00-27-41-0F-95"
    ],
    "os" => {
      "family" => "debian",
      "name" => "Ubuntu",
      "version" => "22.04.4 LTS (Jammy Jellyfish)",
      "type" => "linux",
      "codename" => "jammy",
      "platform" => "ubuntu",
      "kernel" => "5.15.0-97-generic"
    },
    "name" => "luserver22",
    "ip" => [
      [0] "192.168.1.140",
      [1] "fe80::a00:27ff:fe19:8726",
      [2] "192.168.10.1",
      [3] "fe80::a00:27ff:fe41:f95"
    ],
    "id" => "793ca49c79dc4fbf83b36cef40694872",
    "hostname" => "luserver22",
    "containerized" => false,
    "architecture" => "x86_64"
  },
  "type" => "event"
}

```

Hasta este punto sabemos que la información fluye correctamente entre el agente y Logstash. El siguiente paso sería la redirección de estos registros a Elasticsearch en forma de índices, para posteriormente mostrarlos en los tableros de Kibana.

3.4.2 Conexión de Logstash a Elasticsearch:

Lo primero que tenemos que hacer para una comunicación segura es:

- Copiar el certificado de CA autofirmado de Elasticsearch y guardarlo en Logstash.
- Configurar la salida del plugin de salida a elasticsearch para que use el certificado.

Estos pasos no son necesarios si se está usando un certificado de confianza (Emitido por un CA oficial, no autogenerado).

Para realizar la copia del CA se debe acceder al directorio de Elasticsearch "etc/elasticsearch/certs/http_ca.crt. Esta copia debe ubicarse en el directorio de logstash y el usuario debe tener permisos de acceso a este fichero, por lo que podemos cambiar el propietario a root:logstash y establecer los permisos a 755.

Para indicar en la salida de logstash que use este CA debemos indicar:

```
output {
  elasticsearch {
    hosts => ["https://..."] ❶
    cacert => '/etc/logstash/config/certs/ca.crt' ❷
  }
}
```

❶ Note that the `hosts` url must begin with `https`

❷ Path to the Logstash copy of the Elasticsearch certificate

Además, tenemos que configurar algún método de autenticación de usuarios. Logstash necesitará permisos para manejar índices, crearlos, crear y borrar documentos en estos índices.

Estos permisos se pueden otorgar mediante la creación de un rol y un usuario con clave en Kibana o se puede crear una clave api asociada al rol que tiene los permisos adecuados para su gestión.

Se Verá el ejemplo de acceso con usuario y clave para la salida de Logstash a Elasticsearch:

Primero se crea un rol que tenga los permisos necesarios para la escritura de logs en Elasticsearch. Es importante que se permitan los privilegios para los índices que comiencen por "logstash-*", así permitirá la creación de estos índices ya que están restringidos en la configuración de Elasticsearch por seguridad (parámetro: `action.auto_create_index`).

Este es el código para crear un rol con los permisos necesarios. Ejecutamos este código en la terminal de Kibana (dev tool):

```
POST _security/role/logstash_writer
{
  "cluster": ["manage_index_templates", "monitor",
"manage_ilm"],
  "indices": [
    {
      "names": [ "logstash-*" ],
```

```
    "privileges":  
    ["write","create","create_index","manage","manage_ilm"]  
    }  
  ]  
}
```

La ejecución del anterior código en la terminal de Kibana debe crear un rol de usuario para el que, a continuación, se debe crear el usuario que se usará para acceder a Elasticsearch desde Logstash.

```
POST _security/user/logstash_internal  
{  
  "password" : "Admin.123",  
  "roles" : [ "logstash_writer"],  
  "full_name" : "Internal Logstash User"  
}
```

La interfaz nos indica que se ha creado el usuario.

Se crea la configuración de salida hacia Elasticsearch, que debería ser:

```
output {  
  elasticsearch {  
    hosts => ["https://localhost:9200"]  
    cacert => '/etc/logstash/certs/http_ca.cert'  
    ssl => true  
    user => "logstash_internal"  
    password => "Admin.123"  
  }  
}
```

Esta configuración se puede escribir en un fichero de configuración de un pipeline en `"/etc/logstash/conf.d/filebeat_elastic_pipeline.conf"`:


```

input {
  beats {
    port => "5044"
  }
}
#filter {
#}
output {
  elasticsearch {
    hosts => ["https://localhost:9200"]
    index => "logstash-ids"
    ssl => true
    cacert => "/etc/logstash/certs/http_ca.crt"
    user => "logstash_internal"
    password => "Admin.123"
  }
}

```

Finalmente podemos lanzar nuestro logstash con el comando:

```
/usr/share/logstash/bin/logstash -f /etc/logstash/conf.d/filebeat_elastic_pipeline.conf
```

Cuando cargue completamente e indique que está a la escucha en el puerto 5044 se pueden iniciar las pruebas de comunicación para que se registren los logs y podamos observarlos en Kibana.

```

[INFO ] 2024-03-18 00:30:26.000 [[main]-pipeline-manager] jvavapipeline - Pipeline Java execution
[INFO ] 2024-03-18 00:30:26.072 [[main]-pipeline-manager] beats - Starting input listener {:address=>5044}
[INFO ] 2024-03-18 00:30:26.134 [[main]-pipeline-manager] javapipeline - Pipeline started {"pipeline_id":"logstash-intel"}
[INFO ] 2024-03-18 00:30:26.195 [Agent thread] agent - Pipelines running {:count=>1, :running_pipeline_id=>"logstash-intel"}
[INFO ] 2024-03-18 00:30:26.375 [[main]<beats] Server - Starting server on port: 5044

```

Tras algunas pruebas se puede acceder a la interfaz de Kibana y en la sección de “Search → Overview” nos aparece una ventana en la que seleccionaremos “Índices” en el menú de la izquierda y nos debería aparecer el índice creado:

The screenshot shows the Kibana interface for managing Elasticsearch indices. The left sidebar has a menu with 'Indices' highlighted. The main content area is titled 'Elasticsearch indices' and includes a warning banner: 'Enterprise Search has not been configured. The Elastic web crawler is not available without Enterprise Search.' Below this, there are four summary cards: 'Connected ingest methods' (0), 'Incomplete ingest methods' (0), 'Running syncs' (0), and 'Idle syncs' (0). A table titled 'Available indices' shows the 'logstash-ids' index with a health status of 'yellow', 23 documents, and a 'Connected' ingestion status. The table has columns for Index name, Index health, Docs count, Ingestion name, Ingestion method, Ingestion status, and Actions.

Index name	Index health	Docs count	Ingestion name	Ingestion method	Ingestion status	Actions
logstash-ids	yellow	23		API	Connected	

Por último, una vez probada la configuración de logstash y su correcta comunicación con Filebeat y Elasticsearch, se puede iniciar el comando de lanzamiento de Logstash usando múltiples pipelines:

```
# /usr/share/logstash/bin/logstash -f /etc/logstash/pipelines.yml --config.reload.automatic
```

Tras esta ejecución se finalizará el comando, pero el servicio de logstash debe permanecer a la escucha por su puerto 5044.

En ocasiones, puede dar errores en esta ejecución por conflictos en el puerto tras realizar múltiples pruebas. Lo más conveniente es reiniciar el equipo.

Tras el reinicio se puede probar lanzando el comando con el pipelines.yml (que soporta multi pipelines) o, en caso de fallo, se puede optar por realizar un lanzamiento con el único pipeline que realmente se usa en el caso de uso:

```
# /usr/share/logstash/bin/logstash -f /etc/logstash/conf.d/elastic_pipeline.conf --config.reload.automatic
```

3.4.3 Creación de filtros en Logstash (filter)

El siguiente paso consiste en la creación de filtros en Logstash con el uso de grok.

Esta parte puede consultarse directamente en el contenido de la unidad, ya que esta parte es exactamente igual a excepción de algunos cambios en la interfaz de kibana.