

Informe: Implementación de un SIEM Básico con ELK y Suricata

Objetivo: Establecer un sistema SIEM (Security Information and Event Management) funcional utilizando el stack ELK (Elasticsearch, Logstash y Kibana) y Suricata como IDS (Intrusion Detection System).

Entorno: Máquina virtual Linux Ubuntu Desktop 20.04.3 (red en modo puente).

1. Introducción

El stack ELK es una solución robusta para la gestión, análisis y visualización de logs en tiempo real. Al integrarlo con Suricata, podemos transformar datos de eventos de seguridad en información útil para la toma de decisiones. Este informe detalla cada paso del proceso.

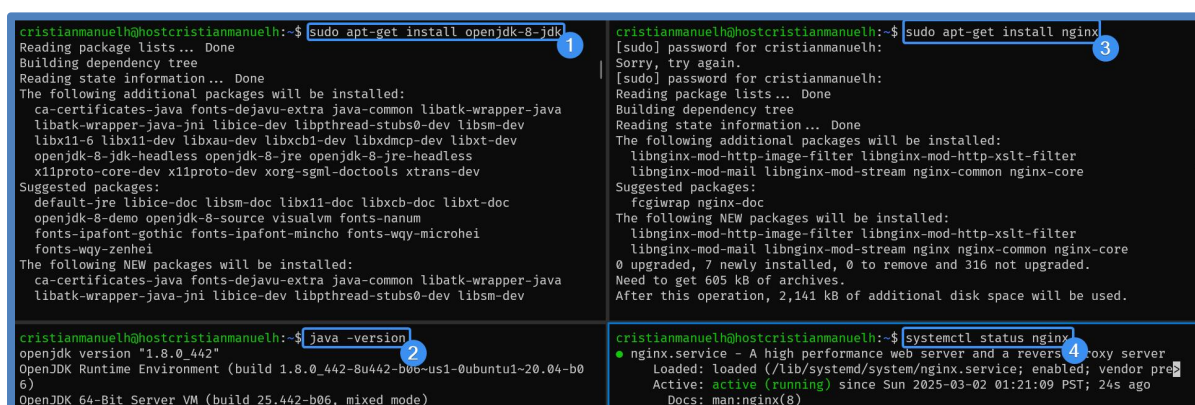
2. Instalación de Componentes Necesarios

2.1. Instalar Java 8 y Nginx

Java 8 es un requisito para el stack ELK. Nginx servirá Kibana.

```
sudo apt-get update
sudo apt-get install openjdk-8-jdk -y
java -version
sudo apt-get install nginx -y
systemctl status nginx
```

Captura:



The screenshot shows a terminal window with four panels of commands and their outputs, numbered 1 through 4. Panel 1 shows the command 'sudo apt-get install openjdk-8-jdk' and its output, including a list of additional packages to be installed. Panel 2 shows the command 'java -version' and its output, 'openjdk version "1.8.0_442"'. Panel 3 shows the command 'sudo apt-get install nginx' and its output, including a list of additional packages to be installed. Panel 4 shows the command 'systemctl status nginx' and its output, showing that the nginx.service is active (running).

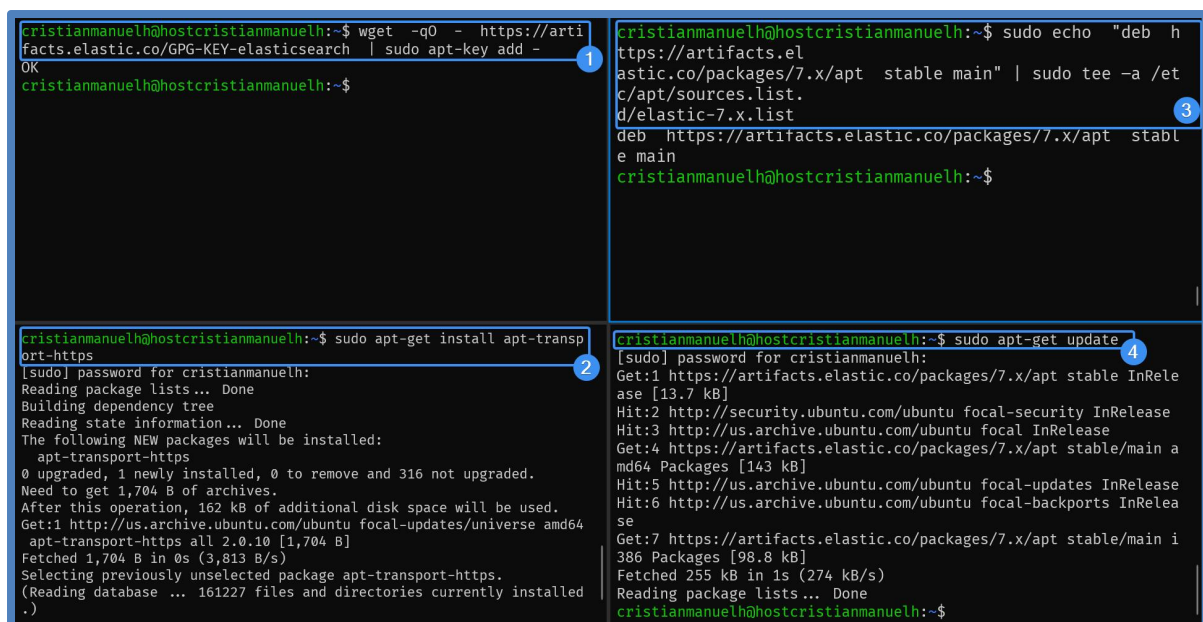
Explicación: Primero, actualizamos la lista de paquetes. Luego instalamos Java 8 y verificamos la versión. Instalamos Nginx para servir Kibana y verificamos que esté en ejecución.

2.2. Añadir el Repositorio de Elastic

Añadimos el repositorio de Elastic para obtener la última versión estable (7.x).

```
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch |
apt-key add -
apt-get install apt-transport-https
echo "deb https://artifacts.elastic.co/packages/7.x/apt stable
main" | tee -a /etc/apt/sources.list.d/elastic-7.x.list
apt-get update
```

Captura:



```
cristianmanuelh@hostcristianmanuelh:~$ wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
OK
cristianmanuelh@hostcristianmanuelh:~$

cristianmanuelh@hostcristianmanuelh:~$ sudo apt-get install apt-transport-https
[sudo] password for cristianmanuelh:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  apt-transport-https
0 upgraded, 1 newly installed, 0 to remove and 316 not upgraded.
Need to get 1,704 B of archives.
After this operation, 162 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 apt-transport-https all 2.0.10 [1,704 B]
Fetched 1,704 B in 0s (3,813 B/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 161227 files and directories currently installed.)
cristianmanuelh@hostcristianmanuelh:~$

cristianmanuelh@hostcristianmanuelh:~$ sudo echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list
deb https://artifacts.elastic.co/packages/7.x/apt stable main
cristianmanuelh@hostcristianmanuelh:~$

cristianmanuelh@hostcristianmanuelh:~$ sudo apt-get update
[sudo] password for cristianmanuelh:
Get:1 https://artifacts.elastic.co/packages/7.x/apt stable InRelease [13.7 kB]
Hit:2 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:4 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 Packages [143 kB]
Hit:5 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:6 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:7 https://artifacts.elastic.co/packages/7.x/apt stable/main i386 Packages [98.8 kB]
Fetched 255 kB in 1s (274 kB/s)
Reading package lists... Done
cristianmanuelh@hostcristianmanuelh:~$
```

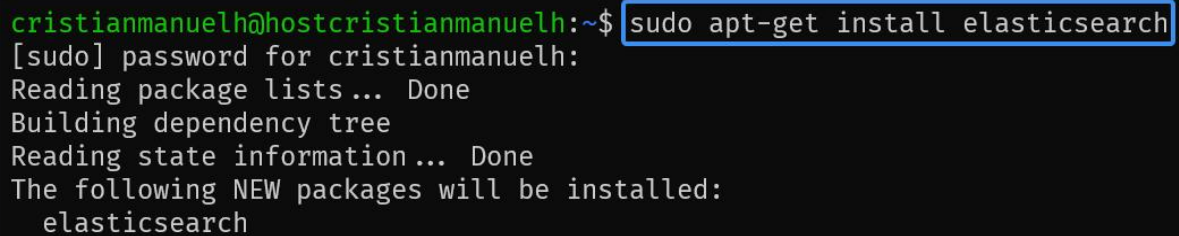
Explicación: Añadimos la clave GPG de Elastic, instalamos `apt-transport-https`, añadimos la línea del repositorio y actualizamos la lista de paquetes.

3. Instalación de Elasticsearch

3.1. Instalar el Paquete Elasticsearch

```
sudo apt-get install elasticsearch
```

Captura:



```
cristianmanuelh@hostcristianmanuelh:~$ sudo apt-get install elasticsearch
[sudo] password for cristianmanuelh:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  elasticsearch
```

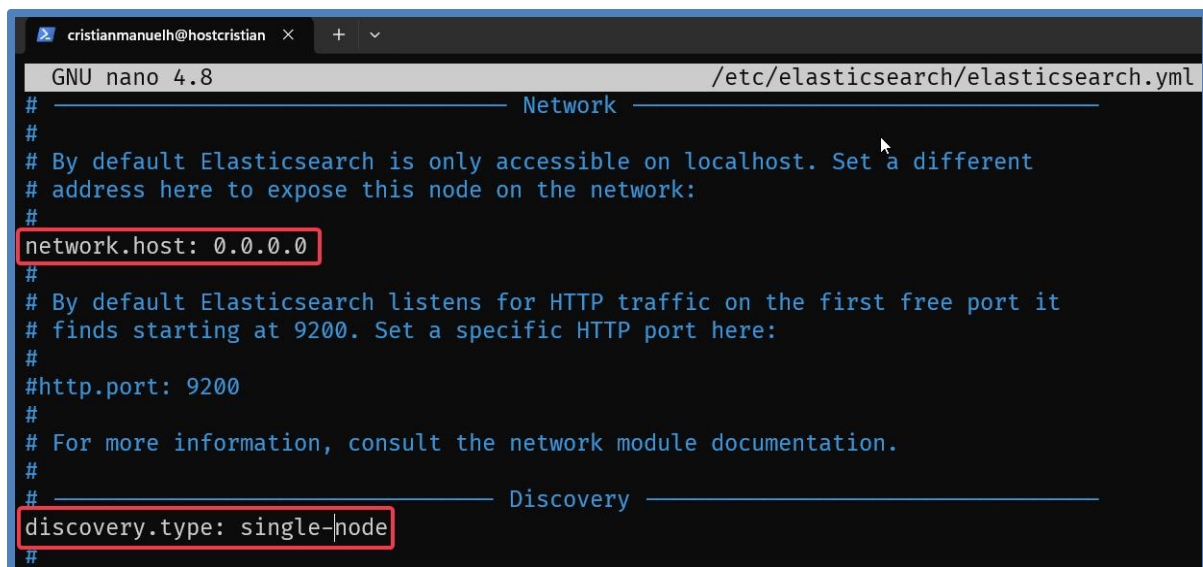
Explicación: Este comando instala el paquete de Elasticsearch desde el repositorio.

3.2. Configurar Elasticsearch

Editamos `/etc/elasticsearch/elasticsearch.yml`:

```
network.host: 0.0.0.0
discovery.type: single-node
```

Captura:



```
cristianmanuelh@hostcristianmanuelh: /etc/elasticsearch/elasticsearch.yml
GNU nano 4.8
# ----- Network -----
#
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
#
network.host: 0.0.0.0
#
# By default Elasticsearch listens for HTTP traffic on the first free port it
# finds starting at 9200. Set a specific HTTP port here:
#
http.port: 9200
#
# For more information, consult the network module documentation.
#
# ----- Discovery -----
discovery.type: single-node
#
```

Explicación: Configuramos Elasticsearch para escuchar en todas las interfaces y establecer el tipo de descubrimiento a `single-node`.

Editamos `/etc/elasticsearch/jvm.options`:

```
-Xms512m  
-Xmx512m
```

Captura:

```
GNU nano 4.8 /etc/elasticsearch/jvm.options  
-Xms512m  
-Xmx512m  
##  
## See https://www.elastic.co/guide/en/elasticsearch/reference/7.17/heap-size.html  
## for more information  
##  
#####
```

Explicación: Reducimos el uso de RAM de la JVM a 512MB para un mejor rendimiento en entornos con recursos limitados.

3.3. Iniciar el Servicio Elasticsearch

```
sudo systemctl enable elasticsearch  
sudo systemctl start elasticsearch  
systemctl status elasticsearch
```

Captura:

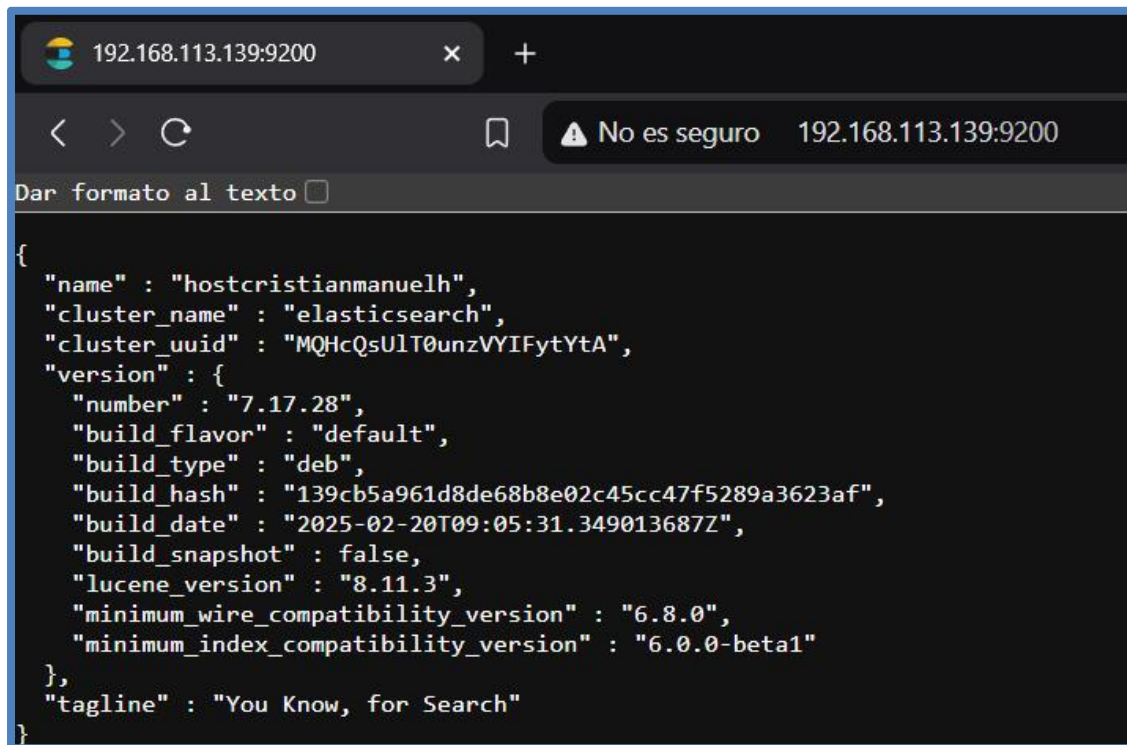
```
cristianmanuelh@hostcristianmanuelh:~$ sudo systemctl enable elasticsearch  
Synchronizing state of elasticsearch.service with SysV service script with /lib/systemd/systemd-sysv-i  
nstall.  
Executing: /lib/systemd/systemd-sysv-install enable elasticsearch  
cristianmanuelh@hostcristianmanuelh:~$ sudo systemctl start elasticsearch  
cristianmanuelh@hostcristianmanuelh:~$ sudo systemctl status elasticsearch  
● elasticsearch.service - Elasticsearch  
   Loaded: loaded (/lib/systemd/system/elasticsearch.service; enabled; vendor preset: enabled)  
   Active: active (running) since Sun 2025-03-02 01:45:43 PST; 31s ago
```

Explicación: Habilitamos Elasticsearch para que se inicie al arrancar el sistema, iniciamos el servicio y verificamos su estado.

3.4. Comprobar el Funcionamiento de Elasticsearch

Visitamos <http://192.168.113.139:9200> en un navegador.

Captura:



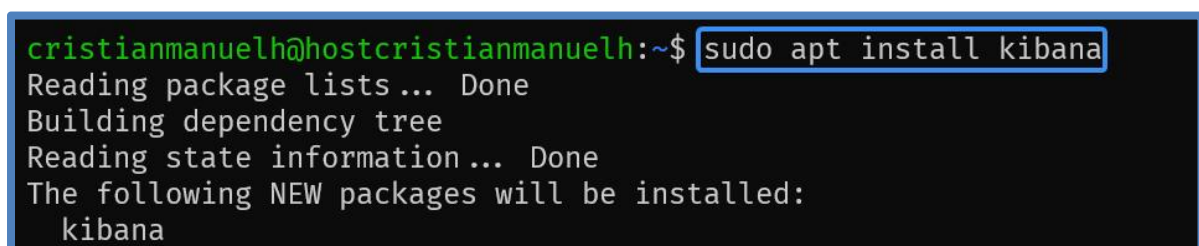
Explicación: La respuesta JSON confirma que Elasticsearch está funcionando correctamente.

4. Instalación de Kibana

4.1. Instalar el Paquete Kibana

```
sudo apt-get install kibana
```

Captura:



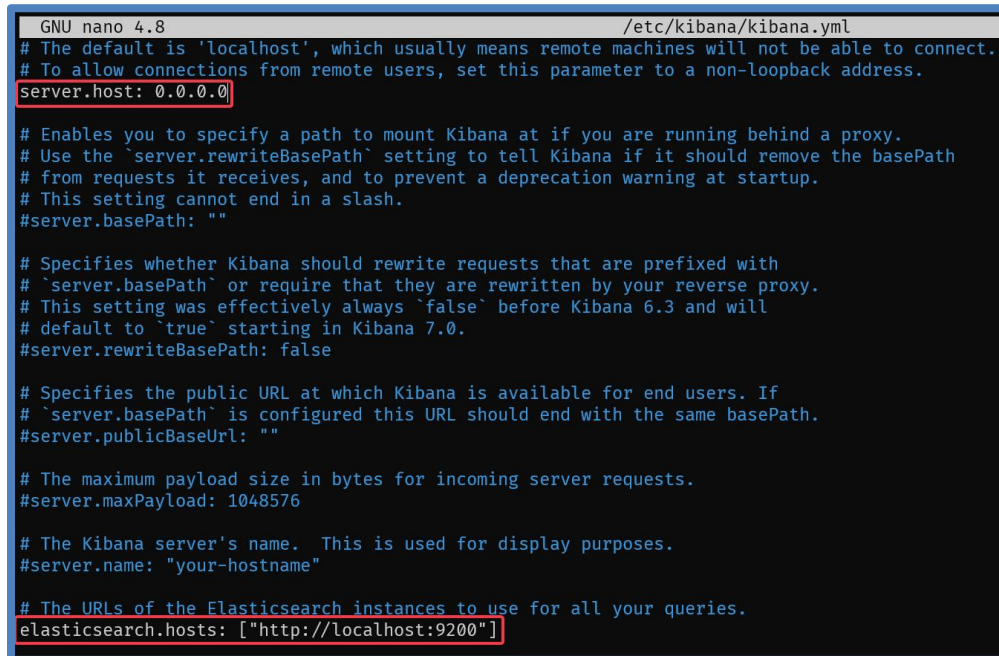
Explicación: Este comando instala el paquete de Kibana desde el repositorio.

4.2. Configurar Kibana

Editamos `/etc/kibana/kibana.yml`:

```
server.host: 0.0.0.0
elasticsearch.hosts: ["http://localhost:9200"]
```

Captura:



```
GNU nano 4.8 /etc/kibana/kibana.yml
# The default is 'localhost', which usually means remote machines will not be able to connect.
# To allow connections from remote users, set this parameter to a non-loopback address.
server.host: 0.0.0.0

# Enables you to specify a path to mount Kibana at if you are running behind a proxy.
# Use the `server.rewriteBasePath` setting to tell Kibana if it should remove the basePath
# from requests it receives, and to prevent a deprecation warning at startup.
# This setting cannot end in a slash.
#server.basePath: ""

# Specifies whether Kibana should rewrite requests that are prefixed with
# `server.basePath` or require that they are rewritten by your reverse proxy.
# This setting was effectively always `false` before Kibana 6.3 and will
# default to `true` starting in Kibana 7.0.
#server.rewriteBasePath: false

# Specifies the public URL at which Kibana is available for end users. If
# `server.basePath` is configured this URL should end with the same basePath.
#server.publicBaseUrl: ""

# The maximum payload size in bytes for incoming server requests.
#server.maxPayload: 1048576

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

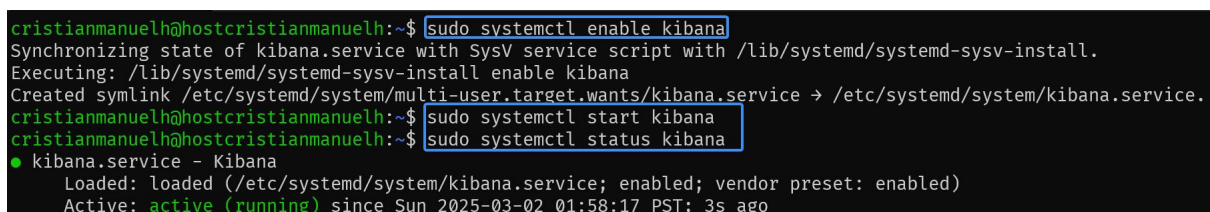
# The URLs of the Elasticsearch instances to use for all your queries.
elasticsearch.hosts: ["http://localhost:9200"]
```

Explicación: Configuramos Kibana para escuchar en todas las interfaces y apuntar a Elasticsearch

4.3. Iniciar el Servicio Kibana

```
sudo systemctl enable kibana
sudo systemctl start kibana
systemctl status kibana
```

Captura Justificada:



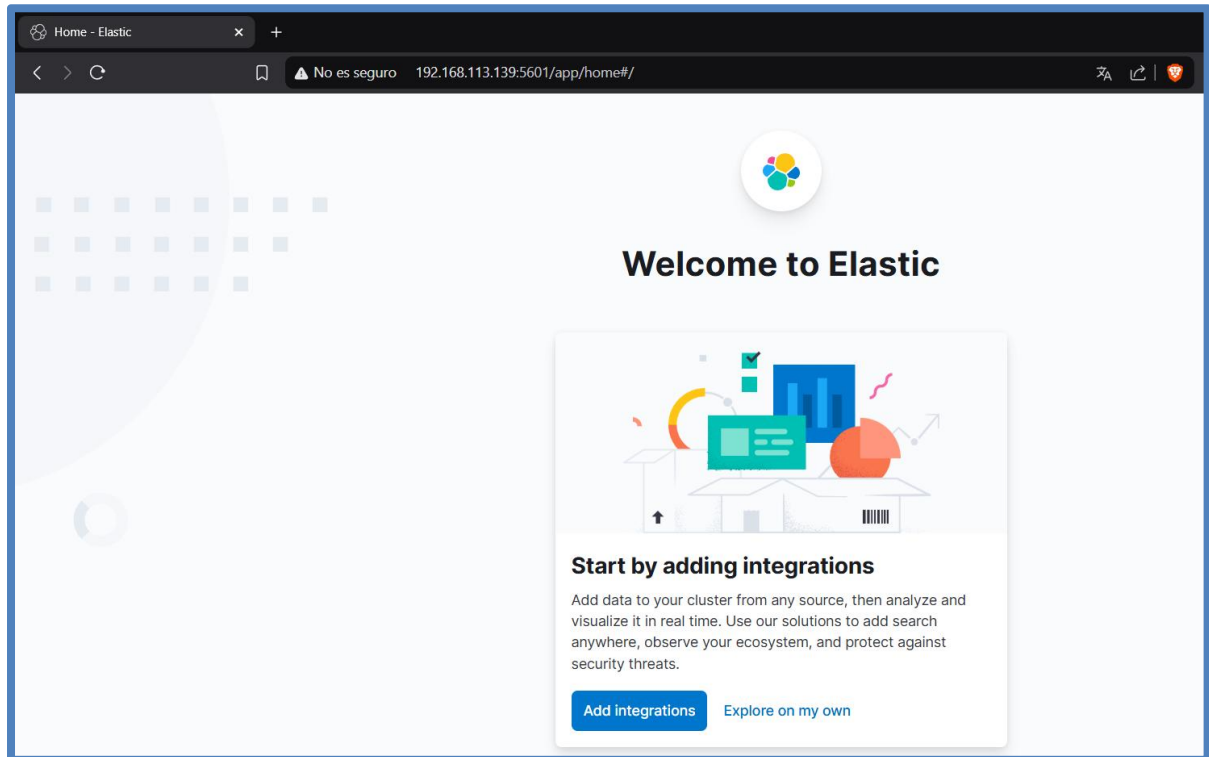
```
cristianmanuelh@hostcristianmanuelh:~$ sudo systemctl enable kibana
Synchronizing state of kibana.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable kibana
Created symlink /etc/systemd/system/multi-user.target.wants/kibana.service → /etc/systemd/system/kibana.service.
cristianmanuelh@hostcristianmanuelh:~$ sudo systemctl start kibana
cristianmanuelh@hostcristianmanuelh:~$ sudo systemctl status kibana
● kibana.service - Kibana
   Loaded: loaded (/etc/systemd/system/kibana.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2025-03-02 01:58:17 PST; 3s ago
```

Explicación: Habilitamos Kibana para que se inicie al arrancar el sistema, iniciamos el servicio y verificamos su estado.

4.4. Comprobar el Funcionamiento de Kibana

Visitamos <http://IPelk:5601> en un navegador.

Captura Justificada:



Explicación: La interfaz de Kibana confirma que está funcionando correctamente.

5. Instalación de Logstash

5.1. Instalar el Paquete Logstash

```
sudo apt-get install logstash
```

Captura:

```
cristianmanuelh@hostcristianmanuelh:~$ sudo apt-get install logstash
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  logstash
```

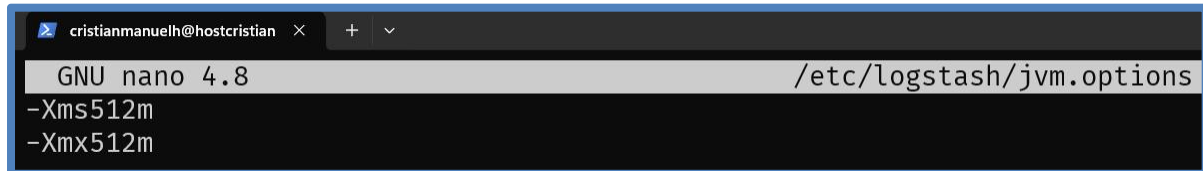
Explicación: Este comando instala el paquete de Logstash desde el repositorio.

5.2. Configurar Logstash

Editamos `/etc/logstash/jvm.options`:

```
-Xms512m  
-Xmx512m
```

Captura:



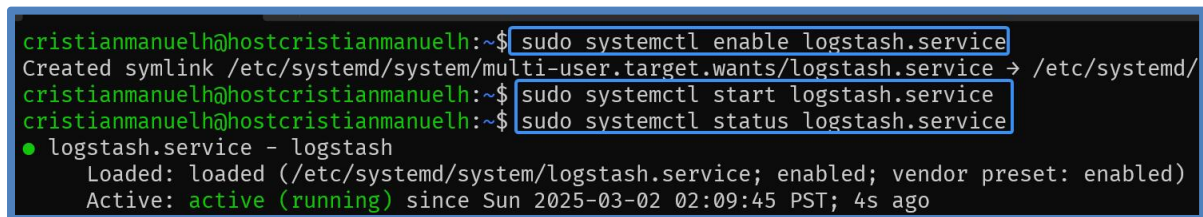
```
cristianmanuelh@hostcristian  GNU nano 4.8  /etc/logstash/jvm.options  
-Xms512m  
-Xmx512m
```

Explicación: Reducimos el uso de RAM de la JVM a 512MB para Logstash.

5.3. Iniciar el Servicio Logstash

```
sudo systemctl enable logstash  
sudo systemctl start logstash  
systemctl status logstash
```

Captura Justificada:



```
cristianmanuelh@hostcristianmanuelh:~$ sudo systemctl enable logstash.service  
Created symlink /etc/systemd/system/multi-user.target.wants/logstash.service → /etc/systemd/  
cristianmanuelh@hostcristianmanuelh:~$ sudo systemctl start logstash.service  
cristianmanuelh@hostcristianmanuelh:~$ sudo systemctl status logstash.service  
● logstash.service - logstash  
   Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor preset: enabled)  
   Active: active (running) since Sun 2025-03-02 02:09:45 PST; 4s ago
```

Explicación: Habilitamos Logstash para que se inicie al arrancar el sistema, iniciamos el servicio y verificamos su estado.

5.4. Comprobar el Funcionamiento del Stack ELK

```
systemctl status elasticsearch kibana logstash nginx
```

Visitamos <http://IPelk:5601/status>

Captura:

```
cristianmanuelh@hostcristianmanuelh:~$ systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
   Loaded: loaded (/lib/systemd/system/elasticsearch.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2025-03-02 01:45:43 PST; 27min ago
     Docs: https://www.elastic.co
   Main PID: 8562 (java)
    Tasks: 63 (limit: 4541)
   Memory: 859.1M
   CGroup: /system.slice/elasticsearch.service
           └─8562 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des
           └─8742 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux
           └─8742 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux

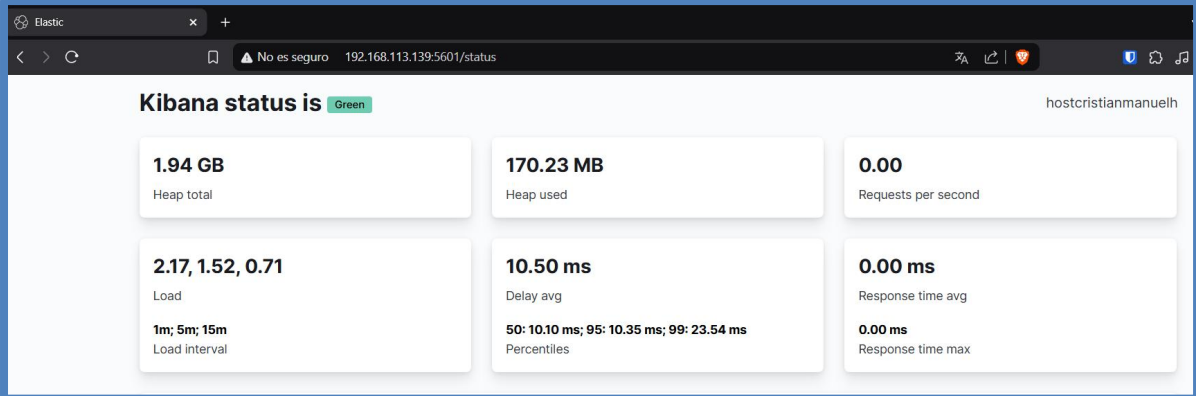
Mar 02 01:45:32 hostcristianmanuelh systemd[1]: Starting Elasticsearch...
Mar 02 01:45:34 hostcristianmanuelh systemd-entrypoint[8562]: Mar 02, 2025
Mar 02 01:45:34 hostcristianmanuelh systemd-entrypoint[8562]: WARNING: COMP
Mar 02 01:45:43 hostcristianmanuelh systemd[1]: Started Elasticsearch.
lines 1-15/15 (END)

cristianmanuelh@hostcristianmanuelh:~$ systemctl status kibana.service
● kibana.service - Kibana
   Loaded: loaded (/etc/systemd/system/kibana.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2025-03-02 01:58:17 PST; 15min ago
     Docs: https://www.elastic.co
   Main PID: 9342 (node)
    Tasks: 11 (limit: 4541)
   Memory: 260.6M
   CGroup: /system.slice/kibana.service
           └─9342 /usr/share/kibana/bin/node --no-deps /usr/share/kibana/bin/node

Mar 02 01:58:17 hostcristianmanuelh systemd[1]: Started Kibana.
Mar 02 01:58:17 hostcristianmanuelh kibana[9342]: Kibana is currently
lines 1-12/12 (END)

cristianmanuelh@hostcristianmanuelh:~$ systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2025-03-02 01:21:09 PST; 52min ago
     Docs: https://nginx.org/en/docs
   Main PID: 1000 (nginx)
    Tasks: 1 (limit: 4541)
   Memory: 1.0M
   CGroup: /system.slice/nginx.service

cristianmanuelh@hostcristianmanuelh:~$ systemctl status logstash.service
● logstash.service - logstash
   Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2025-03-02 02:13:44 PST; 12s ago
     Docs: https://www.elastic.co
   Main PID: 1000 (logstash)
    Tasks: 1 (limit: 4541)
   Memory: 1.0M
   CGroup: /system.slice/logstash.service
```



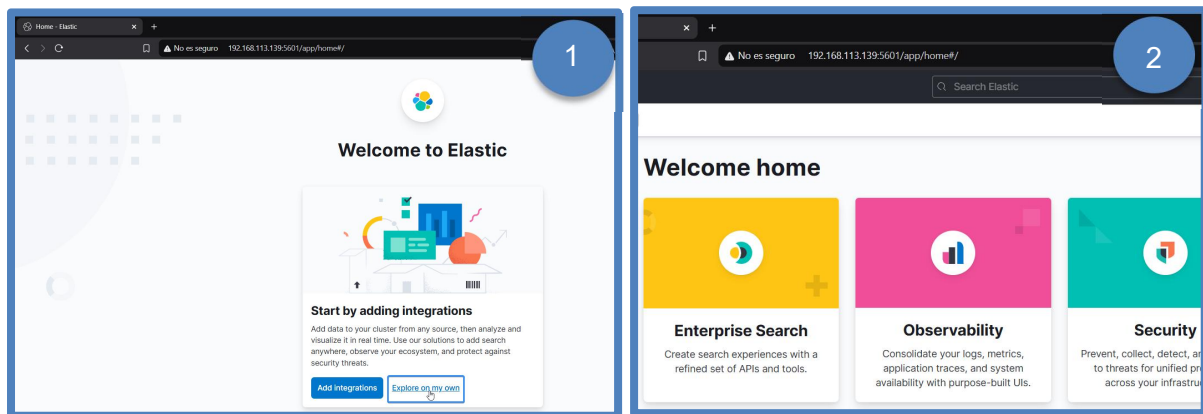
Explicación: Verificamos que todos los servicios estén en ejecución. La página de estado de Kibana muestra información sobre la conexión a Elasticsearch y otros detalles.

6. Configuración de ELK

6.1. Acceder a la Interfaz de Kibana

Visitamos <http://IPelk:5601>. La primera vez, elegimos "Explore on my own".

Captura Justificada:



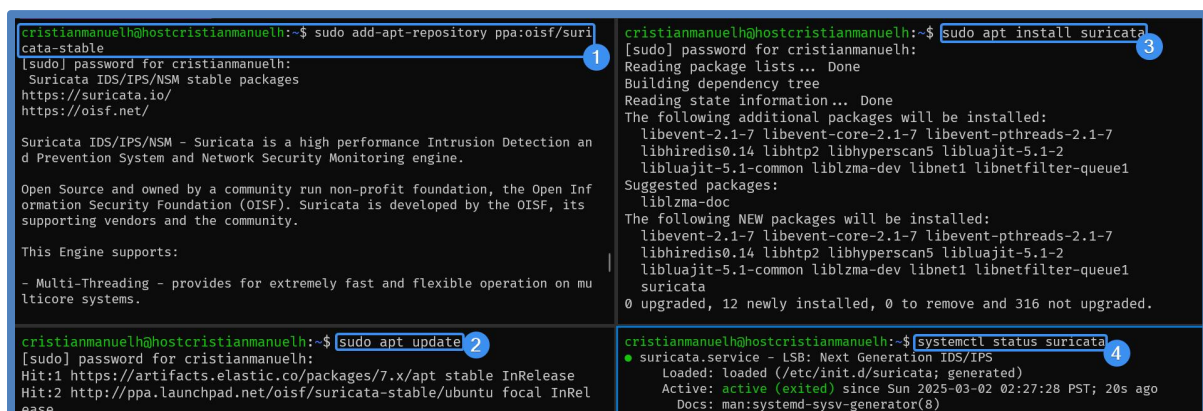
Explicación: Esta opción nos lleva a la interfaz principal de Kibana sin cargar datos predefinidos.

7. Generación de Eventos de Seguridad: Instalación del IDS Suricata

7.1. Añadir el Repositorio e Instalar Suricata

```
sudo add-apt-repository ppa:oisf/suricata-stable
sudo apt update
sudo apt install suricata -y
systemctl status suricata
```

Captura Justificada:



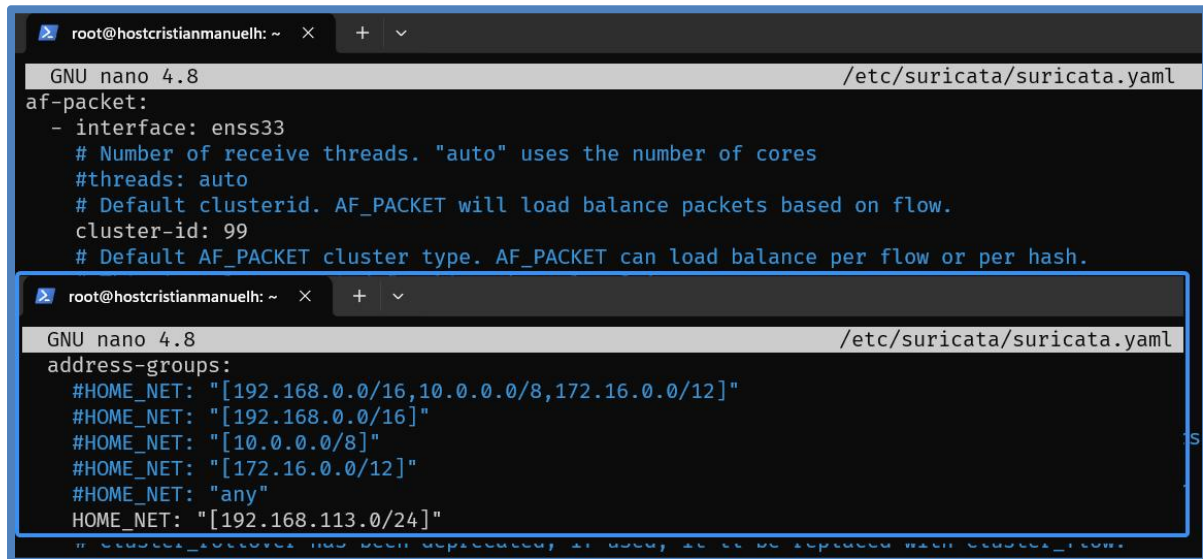
Explicación: Añadimos el repositorio de Suricata y luego instalamos el paquete. Verificamos el estado del servicio.

7.2. Configurar Suricata

Editamos `/etc/suricata/suricata.yaml`:

```
HOME_NET: "[192.168.113.0/24]"
af-packet interface: ens33
```

Captura Justificada:



```
GNU nano 4.8 /etc/suricata/suricata.yaml
af-packet:
- interface: ens33
  # Number of receive threads. "auto" uses the number of cores
  #threads: auto
  # Default clusterid. AF_PACKET will load balance packets based on flow.
  cluster-id: 99
  # Default AF_PACKET cluster type. AF_PACKET can load balance per flow or per hash.

GNU nano 4.8 /etc/suricata/suricata.yaml
address-groups:
#HOME_NET: "[192.168.0.0/16,10.0.0.0/8,172.16.0.0/12]"
#HOME_NET: "[192.168.0.0/16]"
#HOME_NET: "[10.0.0.0/8]"
#HOME_NET: "[172.16.0.0/12]"
#HOME_NET: "any"
HOME_NET: "[192.168.113.0/24]"
```

INFO: Error en el nombre de la tarjeta de red se denominada: **ens33**

Explicación: Establecemos la red local y la interfaz de red para la captura de tráfico.

7.3. Crear Reglas de Suricata

Añadimos las siguientes reglas a `/var/lib/suricata/rules/suricata.rules`:

```
sudo mkdir -p /var/lib/suricata/rules
sudo touch /var/lib/suricata/rules/suricata.rules

echo 'alert icmp any any -> any any (msg:"Ping detectado"; sid:200001;)
alert dns any any -> any 53 (msg:"Petición DNS a google detectada";
dns_query; content:"google"; nocase; sid:200002;)
alert tcp any any -> any 22 (msg:"Conexión SSH detectada";
sid:200003;)' | sudo tee -a /var/lib/suricata/rules/suricata.rules >
/dev/null

sudo suricata -T
```

Captura :

```
cristianmanuelh@hostcristianmanuelh:~$ sudo mkdir -p /var/lib/suricata/rules
cristianmanuelh@hostcristianmanuelh:~$ sudo touch /var/lib/suricata/rules/suricata.rules
cristianmanuelh@hostcristianmanuelh:~$ echo 'alert icmp any any -> any any (msg:"Ping detectado"; sid:200001;)
alert dns any any -> any 53 (msg:"Petición DNS a google detectada"; dns_query; content:"google"; nocase; sid:200002;)
alert tcp any any -> any 22 (msg:"Conexión SSH detectada"; sid:200003;)' | sudo tee -a /var/lib/suricata/rules/suricata.rules > /dev/null
cristianmanuelh@hostcristianmanuelh:~$ sudo suricata -T
i: suricata: This is Suricata version 7.0.8 RELEASE running in SYSTEM mode
i: suricata: Configuration provided was successfully loaded. Exiting.
cristianmanuelh@hostcristianmanuelh:~$
```

Explicación: Creamos el archivo de reglas y añadimos reglas básicas para detectar pings, peticiones DNS a Google y conexiones SSH. Verificamos que la sintaxis sea correcta.

7.4. Reiniciar Suricata

```
sudo systemctl restart suricata
```

Captura Justificada:

```
cristianmanuelh@hostcristianmanuelh:~$ sudo systemctl restart suricata
cristianmanuelh@hostcristianmanuelh:~$
```

Explicación: Reiniciamos el servicio para aplicar las nuevas regla.

7.5. Probar las Reglas de Detección

- ✓ `ping 1.1.1.1`
- ✓ `nslookup www.google.nl`
- ✓ Conectarse por SSH al equipo.

Captura:

```
cristianmanuelh@hostcristianmanuelh:~$ ping 1.1.1.1 -c 4
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=128 time=4.54 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=128 time=3.96 ms
64 bytes from 1.1.1.1: icmp_seq=3 ttl=128 time=4.65 ms
64 bytes from 1.1.1.1: icmp_seq=4 ttl=128 time=4.24 ms

— 1.1.1.1 ping statistics —
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 3.961/4.349/4.653/0.269 ms
cristianmanuelh@hostcristianmanuelh:~$ nslookup www.google.nl
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   www.google.nl
Address: 142.250.200.131
Name:   www.google.nl
Address: 2a00:1450:4003:80f::2003

PS C:\Users\cristian> ssh cristianmanuelh@192.168.113.139
cristianmanuelh@192.168.113.139's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-131-generic x86_64)
```

Explicación: Ejecutamos estos comandos para generar tráfico que active las reglas de Suricata.

7.6. Comprobar las Alertas de Suricata

```
tail -f /var/log/suricata/fast.log
```

Captura:

```
cristianmanuelh@hostcristianmanuelh:~$ tail -f /var/log/suricata/fast.log
03/02/2025-03:44:03.223937  [**] [1:200003:0] Conexion SSH detectada [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.113.1:32586 → 192.168.113.13
9:22
03/02/2025-03:44:03.269992  [**] [1:200003:0] Conexion SSH detectada [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.113.1:44300 → 192.168.113.13
9:22
03/02/2025-03:44:17.978823  [**] [1:200001:0] Ping detectado [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.113.139:8 → 1.1.1.1:0
03/02/2025-03:44:17.983615  [**] [1:200001:0] Ping detectado [**] [Classification: (null)] [Priority: 3] {ICMP} 1.1.1.1:0 → 192.168.113.139:0
03/02/2025-03:44:30.956265  [**] [1:200002:0] Peticion DNS a google detectada [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.113.139:44458 → 192
.168.113.2:53
03/02/2025-03:44:30.988768  [**] [1:200002:0] Peticion DNS a google detectada [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.113.139:50970 → 192
.168.113.2:53
```

Explicación: Verificamos que las alertas se estén registrando en el archivo de log de Suricata.

8. Ingesta de Datos en ELK: Configuración de Logstash

8.1. Analizar el Formato del Log de Suricata

Abrimos `/var/log/suricata/fast.log` y copiamos una línea de ejemplo.

```
03/02/2025-03:44:17.983615  [**] [1:200001:0] Ping detectado [**]
[Classification: (null)] [Priority: 3] {ICMP} 1.1.1.1:0 ->
192.168.113.139:0
```

8.2. Diseñar un Patrón Grok

Utilizamos el Grok Debugger en Kibana (**Management -> Dev Tools -> Grok Debugger**) para diseñar un patrón Grok.

Ejemplo de patrón:

```
%{DATE:fecha}%{TIME:hora}.....%{WORD:sid}....%{GREEDYDATA:evento}.
%{WORD:protocolo}}.%{IP:srcip}:%{INT:srcport}.>.%{IP:dstip}:%{INT:dstp
ort}
```

Captura:



Explicación: Podemos ver que el diseño del patrón Grok, nos divide correctamente el log.

8.3. Crear el Archivo de Configuración de Logstash

Creamos el archivo `/etc/logstash/conf.d/eventos.conf`:

```
input {
  file {
    type => "suricata"
    path => "/var/log/suricata/fast.log"
    start_position => beginning
  }
}
filter {
  grok {
    match => { "
=>"%{DATE:fecha}%{TIME:hora}.....%{WORD:sid}...%{GREEDYDATA:event
o}..%{WORD:protocolo}}.%{IP:srcip}:%{INT:srcport}.->.%{IP:dstip}:%{INT:d
stport}"
    }
  }
}
output {
  elasticsearch {
    hosts => "http://localhost:9200"
    index => "eventos"
  }
}
```


Captura:

```
GNU nano 4.8 /etc/logstash/conf.d/eventos.conf
input {
  file {
    type => "suricata"
    path => "/var/log/suricata/fast.log"
    start_position => beginning
  }
}
filter {
  grok {
    match => { "message" => "%{DATE:fecha}%{TIME:hora}.....%{WORD:sid}....%{GREEDYDATA:evento}"
  }
}
output {
  elasticsearch {
    hosts => "http://localhost:9200"
    index => "eventos"
  }
}
```

Explicación: Muestra el archivo de configuración de Logstash con el input, filter y output definidos.

8.4. Asignar Permisos a los Archivos de Log

```
sudo chmod 777 /var/log/suricata/fast.log
```

Captura Justificada:

```
cristianmanuelh@hostcristianmanuelh:~$ sudo chmod 777 /var/log/suricata/fast.log
cristianmanuelh@hostcristianmanuelh:~$
```

Explicación: Establecemos los permisos sobre el directorio `/var/log/suricata/fast.log`.

8.5. Reiniciar Logstash y Kibana

```
sudo systemctl restart logstash
sudo systemctl restart kibana
```

Captura Justificada:

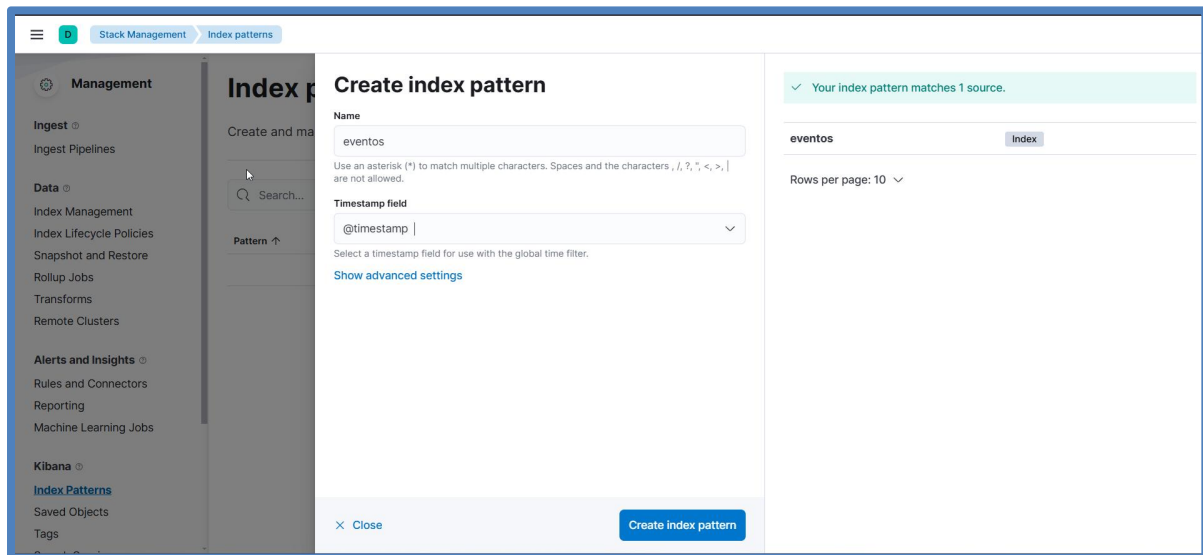
```
cristianmanuelh@hostcristianmanuelh:~$ sudo systemctl restart logstash
cristianmanuelh@hostcristianmanuelh:~$ sudo systemctl restart kibana
```

Explicación: Reiniciamos los servicios de logstash y kibana.

9. Añadir Eventos a Elasticsearch

Accedemos en la web de Kibana al menú **Management->Stack Management->Index Pattern** y pulsamos el botón **Create index pattern** y creamos el índice con nombre **eventos** y campo **@timestamp**.

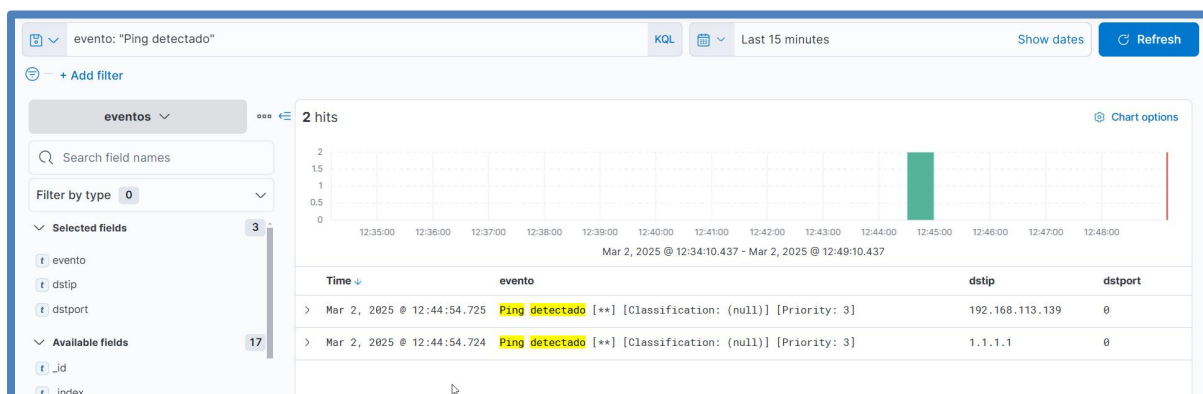
Captura:



10. Consulta de Eventos en el SIEM

Visitamos <http://IPElk:5601> y accedemos al menú **Analytics->Discover**.

Captura:



Explicación: Filtramos los eventos mostrados en la tabla usando la barra de búsqueda en el campo 'evento' contenga la cadena de texto 'Ping detectado':

11. Mostrar Gráficas de Eventos en el SIEM

Para **mostrar gráficas** de los eventos en un panel de control (dashboard), accederemos al menú **Analytics->Dashboard** y pulsaremos el botón **Create new dashboard**.

Luego pulsaremos en el botón **Create visualization** para crear algunos de los gráficos más comunes.

Para cada gráfico elegido debemos indicarle el index (la fuente de datos, que en este caso será eventos) y le agregaremos el campo o campos de los eventos que queremos mostrar en el gráfico.

Bar vertical (Histograma): Gráfica de barras verticales

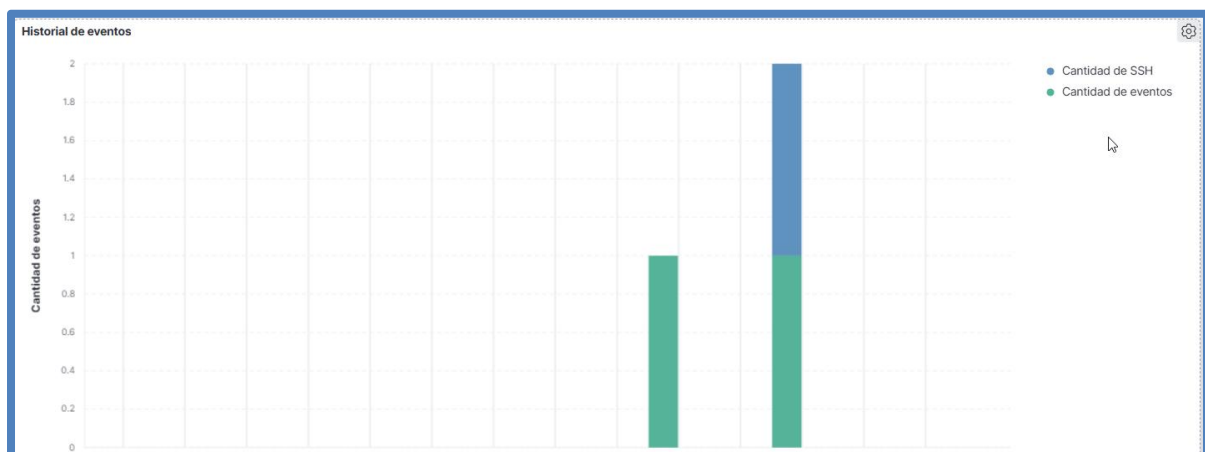
Title: Historial de eventos

- ✓ **Horizontal axis:**
 - ◆ **Date histogram:** @timestamp
- ✓ **Vertical axis:**
 - ◆ **Select a function:** Count
 - ◆ **Select a field:** Records
 - ◆ **Display name:** Cantidad de eventos

Modificar la gráfica para añadir otra barra que muestre el total de eventos de tipo '**Conexion SSH detectada**' para poder compararlos frente al total de eventos:

- ✓ **Vertical axis:**
 - ◆ **Select a function:** Count
 - ◆ **Select a field:** Records
 - ◆ **Add advanced options:**
 - **Filter by:** "Conexion SSH detectada"
 - ◆ **Display name:** Cantidad de conexiones SSH

Captura:



Line (Línea): Gráfica de líneas verticales.

Title: Evolución de los pings detectados

- ✓ Horizontal axis:
 - ◆ Date histogram: @timestamp
- ✓ Vertical axis:
 - ◆ Select a function: Count
 - ◆ Select a field: Records
 - ◆ Add advanced options:
 - Filter by: "Ping detectado"
 - ◆ Display name: Cantidad de pings

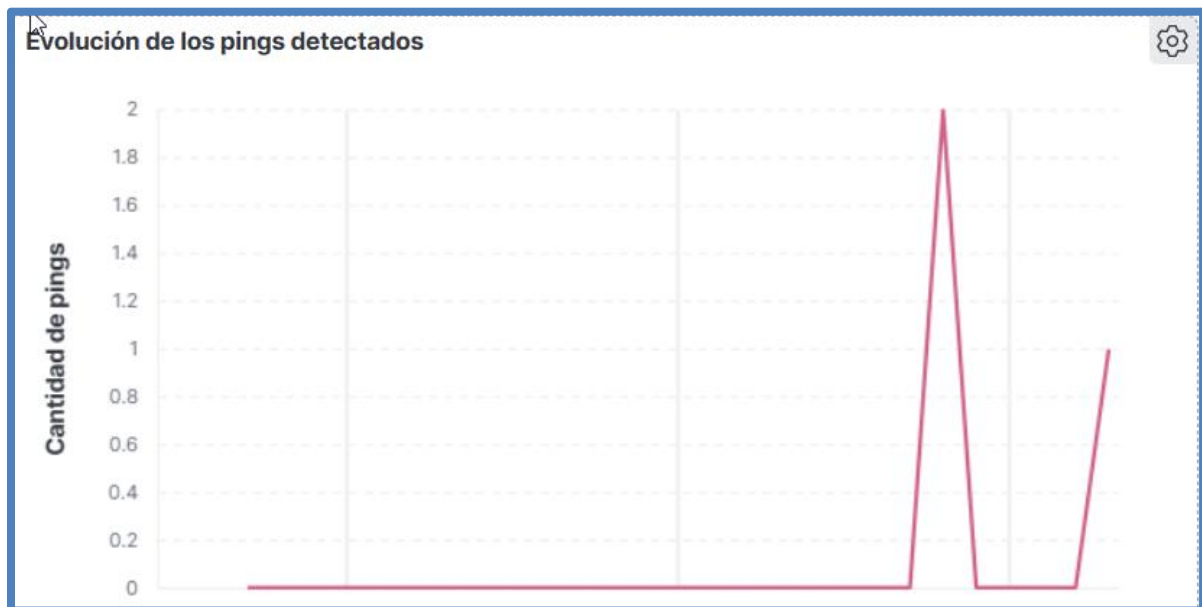


Table (Tabla): Tabla que muestra la cantidad de eventos recibidos en el SIEM por tipo.

Title: Total de eventos

- ✓ Rows:
 - ◆ Select a function: Top values
 - Select a field: evento.keyword
 - Number of values: 5
- ✓ Metrics:
 - ◆ Select a function: Count
 - ◆ Select a field: Records

Total de eventos	
Evento detectado	Cantidad
Conexion SSH detectada [**] [Classifi...	3
Peticion DNS a google detectada [**] [...	2
Ping detectado [**] [Classification: (nu...	2

Metric (Métrica): Muestra un contador de eventos (registros).

Title: Cantidad total de eventos

✓ **Metric:**

◆ **Select a function:** Count

■ **Select a field:** Records

■ **Display name:** Cantidad de eventos



Title: Cantidad de conexiones SSH

✓ **Metric:**

◆ **Select a function:** Count

■ **Select a field:** Records

■ **Add advanced options:**

❖ **Filter by:** "Conexion SSH detectada"

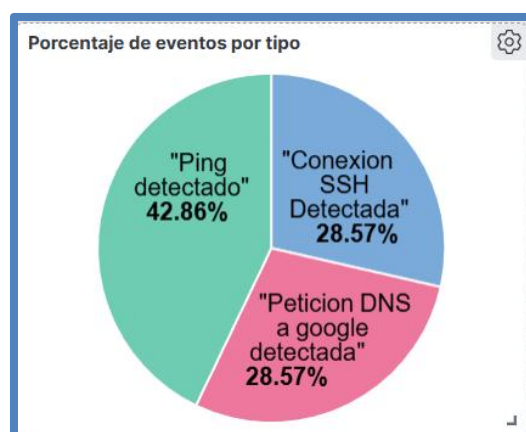
◆ **Display name:** Cantidad de conexiones SSH



Donut / Pie (Donut / Circular): Gráfica circular para mostrar cantidades o porcentajes.

Title: Porcentaje de eventos por tipo

- ✓ **Slice by:**
 - ◆ **Filters:** "Ping detectado"
 - ◆ **Filters:** "Conexion SSH detectada"
 - ◆ **Filters:** "Petición DNS a google detectada"
- ✓ **Size by:**
 - ◆ **Select function:** Count
 - ◆ **Select field:** Records



12. Pruebas y Refresco

Ejecutamos comandos para generar eventos y pulsamos el botón **Refresh** en Kibana.



13. Conclusiones

Este informe ha detallado el proceso de implementación de un SIEM básico utilizando el stack ELK y Suricata. Este sistema permite la captura, análisis y visualización de eventos de seguridad en tiempo real, mejorando la capacidad de detección de incidentes y la toma de decisiones. La implementación de políticas de seguridad mejorará de forma sustancial la seguridad de su entorno.