



Universitat
Oberta
de Catalunya

UAB
Universitat Autònoma
de Barcelona



UNIVERSITAT ROVIRA I VIRGILI

Trabajo final de máster

Proyecto: MiShodan

Memoria del proyecto fin del Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones (*MISTIC*), realizado por <mi nombre> y dirigido por <mi tutor>

Madrid, 4 de junio de 2020

Resumen

Siempre que una empresa contrata a un auditor en su plantilla con el objetivo de mantener sus sistemas tanto externos como internos seguros, el auditor solicitará una relación de activos y su criticidad para la empresa, así como la infraestructura de red en donde aparezcan todos los sistemas, ya sean servicios web, bases de datos, sistemas de seguridad, etc.

El problema empieza cuando la empresa no sabe dar respuesta a esa pregunta, ya que a partir de ahí el auditor tendrá que encontrarlos, principalmente entrevistando a los distintos miembros de la empresa. Esta tarea puede llegar a ser un ejercicio sencillo, arduo o extenuante dependiendo de la empresa en donde se desarrolle la actividad.

Pero, si a ese enfoque se le añade una labor de caja negra de localización de direcciones IP activas dentro de la empresa, teniendo en cuenta que cada IP se corresponde con un activo de la empresa, el tiempo de investigación se reduciría.

En la búsqueda de esas IPs, el auditor empezarán sus pesquisas desde el equipo que se le haya asignado dentro de la empresa mirando la IP de su equipo y las IPs de los servidores DNS para hacerse una idea de los rangos de IPs manejados por la empresa. A partir de ahí, empezará a trabajar con distintas herramientas para el descubrimiento de activos, una de esas herramientas puede ser la archiconocida: nmap.

Sin embargo esa tarea necesita un tiempo que el auditor puede no tener debido a la cantidad de entrevistas a realizar.

Por dicho motivo, sería necesario automatizar dicho proceso, lo que implicaría una reducción del tiempo de investigación que el auditor emplearía.
Ese proceso de automatización es lo que se describe en este documento que está usted leyendo.

Espero que sea de su agrado.



Overview

When one company hires a auditor inside its staff with the objective to keep theirs computer systems such internal as external safe, the auditor asks for the assets relationship and their criticality for the company, just like the network structure where every systems are, such as web service, data base, security system, etc.

The problem starts when the company doesn't know to response that question, since there the auditor will have to find it mainly interviewing all of the members of the staff. This task could became a easy,a difficult or exhausting task, depending the company where the auditor is.

But if to the previous point of view we adds a black box task of localization the IP directions inside the company, considering that each IP is a asset of the company, the investigation time will be less

For the search of those IPs, the auditor will start her/his search from the computer inside the company, seeing her/his IP and the IPs of the DNS servers. With these data the auditor could know the ranges of IPs of the company. From there, the auditor will start to work with several tools for discovering the assets, one of these tools can be: nmap

However, that task needs a time what the auditor couldn't have due to the number of interviews to be conducted

That's why, it will be neccesary to automative this task, this will get to have less time of investigation by auditor

That task of automation is described in this document.

I hope it's to your liking

ÍNDICE

1. INTRODUCCIÓN.....	9
1.1. Motivación del proyecto.....	9
1.2. Objetivos	9
1.3. Estado del arte.....	9
1.4. Viabilidad	9
1.5. Presupuesto	11
1.6. Descripción del trabajo a realizar.....	11
1.7. Flujos de información del proyecto.....	15
2. FASE 1: SONDA DE ENUMERACIÓN.....	16
3. FASE 2: DESARROLLO DE LA APLICACIÓN.....	20
3.1. Shodan.sh.....	21
3.2. ParseoShodan.sh	24
3.3. ControlEstado.sh	27
3.4. Creación de un demonio propio	28
3.5. Creación de la carpeta: Sonda	29
4. FASE 3: SERVIDOR WEB, BBDD Y PROXY-WEB	30
5. FUTUROS PASOS.....	34
6. VALORACIONES PERSONALES	35
ANEXO I: PROCESO DE INSTALACIÓN DE LA SONDA DE ENUMERACIÓN	36
AI.1. Creación de la máquina virtual.....	36
AI.2. Instalación del sistema operativo	42
AI.3. Instalación de la herramienta: NMAP	52
AI.4. Instalación de la herramienta: Logstash	55
AI.5. Bastionado del servicio: SSH	58
ANEXO II: CODIGOS COMPLETOS	61
AII.1. Shodan.sh.....	61
AII.2. ParseoShodan.sh	64

AIII.3. ControlEstado.sh	66
ANEXO III: PROCESO DE INSTALACIÓN DE LOS SERVIDORES WEB, BBDD Y PROXY-WEB	68
AIII.1. Instalación de ElasticSearch	68
AIII.2. Instalación de Kibana	72
AIII.3. Instalación de Nginx	75
BIBLIOGRAFÍA	77

TABLA DE ILUSTRACIONES

Ilustración 1: Plan de trabajo - Tareas	14
Ilustración 2: Plan de trabajo - Tiempos	14
Ilustración 3: Flujos de comunicaciones del proyecto	15
Ilustración 4: Parte de información recaba.....	16
Ilustración 5: Configuración de Logstash.....	17
Ilustración 6: Comprobación básica de las comunicaciones Logstash.....	18
Ilustración 7: Recepción de las comunicaciones básicas de Logstash	18
Ilustración 8: Comprobación recepción del contenido del archivo "cit_resources.csv".	19
Ilustración 9: Shodan.sh - Variables.....	21
Ilustración 10: Shodan.sh – Control del formato de la última IP analizada	21
Ilustración 11: Shodan.sh – Enumeración de todas las IPs de un rango.....	22
Ilustración 12: Shodan.sh - Uso de la memoria.....	23
Ilustración 13: Shodan.sh - Control de procesos de nmap.....	23
Ilustración 14: ParseoShodan.sh - Variables	24
Ilustración 15: ParseoShodan.sh - Parseo de la información	26
Ilustración 16: ParseoShodan.sh - Tiempo de espera	26
Ilustración 17: Lanzador-Shodan.sh.....	28
Ilustración 18: Rc.local.....	28
Ilustración 19: Creada carpeta: Sonda.....	29
Ilustración 20: Contenido de la carpeta: Sonda.....	29
Ilustración 21: Configuración utilizada en Nginx.....	31
Ilustración 22: Proxy-web tras el puerto 80 del servidor del proyecto.....	32
Ilustración 23: Creación MV - Paso 1	36
Ilustración 24: Creación MV - Paso 2.....	37
Ilustración 25: Creación MV - Paso 3.....	38
Ilustración 26: Creación MV - Paso 4.....	39
Ilustración 27: Creación MV - Paso 5.....	40
Ilustración 28: Creación MV - Paso 6.....	41
Ilustración 29: Instalación sistema operativo - Paso 1	42
Ilustración 30: Instalación sistema operativo - Paso 2	43
Ilustración 31: Instalación sistema operativo - Paso 3	43
Ilustración 32: Instalación sistema operativo - Paso 4	44
Ilustración 33: Instalación sistema operativo - Paso 5	45
Ilustración 34: Instalación sistema operativo - Paso 6	46
Ilustración 35: Instalación sistema operativo - Paso 7	47
Ilustración 36: Instalación sistema operativo - Paso 8	47
Ilustración 37: Instalación sistema operativo - Paso 9	48
Ilustración 38: Instalación sistema operativo - Paso 10	49
Ilustración 39: Instalación sistema operativo - Paso 11	49
Ilustración 40: Instalación sistema operativo - Paso 12	49
Ilustración 41: Instalación sistema operativo - Paso 13	50

Ilustración 42: Instalación sistema operativo finalizada	51
Ilustración 43: Nmap - Escalamos privilegios	52
Ilustración 44: Nmap - Actualizamos repositorios	52
Ilustración 45: Nmap - Instalamos nmap	52
Ilustración 46: Nmap - Instalación completada.....	53
Ilustración 47: Nmap - Archivo de catalogación de servicios de nmap.....	53
Ilustración 48: Logstash - Instalación JAVA.....	55
Ilustración 49: Logstash - Instalación JAVA completada	55
Ilustración 50: Logstash - Descarga clave pública acceso a repositorio Logstash.....	55
Ilustración 51: Logstash - Inclusión clave pública en nuestro sistema	55
Ilustración 52: Logstash - Instalación apt-transport-https	56
Ilustración 53: Logstash - Incluimos los repositorios de logstash en nuestro fichero de fuentes.....	56
Ilustración 54: Logstash - Instalación Logstash.....	56
Ilustración 55: LogStash - Lanzador-LogsStash.sh	57
Ilustración 56: LogStash - rc.local.....	57
Ilustración 57: SSH - Ubicación de los archivos de configuración	58
Ilustración 58: SSH - Banner	59
Ilustración 59: SSH - Definimos los usuarios que pueden usar el servicio SSH	59
Ilustración 60: SSH - Tiempo máximo para acceder	59
Ilustración 61: SSH - Máximo número de intentos de acceso.....	60
Ilustración 62: SSH - Número máximo de sesiones concurrentes	60
Ilustración 63: SSH - No permitimos validación por parte de ROOT	60
Ilustración 64: Códigos - Shodan.sh - 1	62
Ilustración 65: Códigos - Shodan.sh -2.....	63
Ilustración 66: Códigos - ParseoShodan.sh.....	65
Ilustración 67: Códigos - ControlEstado.sh	67
Ilustración 68: ElasticSearch - Instalación de Java.....	68
Ilustración 69: ElasticSearch - Tras la llamada de instalación	68
Ilustración 70: ElasticSearch - Instalación de ElasticSearch.....	69
Ilustración 71: ElasticSearch - Configuración de ElasticSearch - Parte 1	69
Ilustración 72: ElasticSearch - Configuración de ElasticSearch - Parte 2.....	70
Ilustración 73: ElasticSearch - Iniciando el servicio.....	70
Ilustración 74: ElasticSearch - Lanzador-EntornoELK.sh	70
Ilustración 75: ElasticSearch - rc.local	71
Ilustración 76: Kibana - Instalación de Kibana.....	72
Ilustración 77: Kibana - Configuración del Kibana	73
Ilustración 78: Kibana - Iniciando el servicio.....	73
Ilustración 79: Kibana - Lanzador-EntornoELK.sh	73
Ilustración 80: Kibana - rc.local	74
Ilustración 81: Nginx - Instalación de Nginx y apache2-utils	75
Ilustración 82: Nginx - Configuración utilizada en Nginx.....	76



1. INTRODUCCIÓN

1.1. Motivación del proyecto

La motivación realmente es, tal y como ya se ha manifestado anteriormente, ahorrarme tiempo de trabajo, ya que la situación planteada en el resumen es una situación personal vivida en múltiples clientes.

1.2. Objetivos

Los principales objetivos del Trabajo Final de Máster (TFM) son los siguientes:

- Generar una solución que permita realizar una enumeración de todos los activos que posee una empresa y los servicios que ese activo posee, con lo que poder establecer un listado de activos y delimitar la criticidad de los mismos.
- Generar una solución que sea poco probable que detecten los posibles elementos de seguridad perimetral que tenga la red donde se ejecute la sonda.
- Generar una solución que sea fácilmente desplegable en cualquier entorno
- Generar una solución económica
- Generar una solución sencilla de mantener
- Generar una solución que trabaje 24*7*366

1.3. Estado del arte

Es verdad que existen múltiples herramientas de enumeración/escaneo de activos/IP, tanto de pago como open source, también es verdad que existen múltiples herramientas/entornos parecidos al seleccionado para este proyecto. Pero no se ha podido encontrar ninguna herramienta ni de pago ni open source que cumplan los objetivos marcados en este trabajo.

Por dicho motivo, la solución que se propone en este proyecto es una solución novedosa.

1.4. Viabilidad

La viabilidad del proyecto es totalmente asumible, principalmente por el hecho de que nos encontramos ante una solución generada en parte por software propio y por otra parte por el uso de herramientas de carácter open source.



1.5. Presupuesto

Anteriormente se ha comentado que se busca una solución económica, por lo que se han buscado herramientas de tipo: open source. Por lo que el gasto derivado de este tipo de herramientas es de: 0 euros.

Los siguientes elementos son open-source:

- 1.- Sistema operativo Linux 18.04.4
- 2.- La herramienta: nmap
- 3.- La herramienta: ELK (Elasticsearch Logstash Kibana)
- 4.- El desarrollo realizado para ésta solución

Por otro lado, necesitaremos:

1.- Máquina Entorno de gestión de máquinas virtuales, en mi caso he utilizado la solución: VMWare WorkStation Pro 15.5, con un coste de: 331,95 euros por un año [\[0\]](#)

Es posible sustituir el gestor de máquinas virtuales por el gestor llamado: VirtualBox, generador por IBM. Este gestor es gratuito.

2.- Máquina que hospedará las máquinas virtuales. El coste de dicha máquina no debería ser excesivo, ya que las máquinas virtuales a crear no suponen muchas necesidades a nivel de hardware.

1.6. Descripción del trabajo a realizar

A continuación, se presentan los diferentes hitos y una breve descripción de lo esperado:

NOTA: El software de virtualización necesario para realizar este proyecto será: **VMWare**

El TFM, se dividirá en tres fases de trabajo.

A. **Primera fase.** Instalación de la máquina virtual que trabajará como una sonda de enumeración.

Tiempo de trabajo: del 12/03/2020 al 30/03/2020

Dicha sonda permitirá recabar direccionamientos IPs que tienen algún servicio/puerto abierto

Para tener la sonda perfectamente instalada se seguirán los siguientes hitos:

1. Se utilizará un sistema operativo Ubuntu Linux 18.04 sin interfaz gráfica [1], por su poca necesidad de recursos y su carácter gratuito.

NOTA: El sistema que se ha utilizado tiene 2Gb de RAM, 2 procesadores y el espacio de disco duro que define VMWare por defecto (20 Gb)

2. Se abrirá el puerto 22/TCP para la administración remota.

NOTA: Adicionalmente este puerto será bastionado

Tiempo de trabajo (2 puntos anteriores): 12/03/2020 al 17/03/2020

3. Se instalará la herramienta **Nmap** [2], con la cuál lanzaremos los escaneos/descubrimientos de IPs, sistemas con algún servicio/puerto abierto.

NOTA: Se utilizará el fichero de categorización de dicha herramienta para catalogar los puertos/servicios encontrados.

Tiempo de trabajo: 18/03/2020 al 19/03/2020

4. Se instalará la herramienta: **logstash** [3], que permitirá el envío de la información recopilada por la sonda hacia el servidor de base de datos/web. Para que logstash funcione correctamente se debe instalar el aplicativo: **java**

NOTA: La herramienta **logstash** se encuentra dentro de la solución ELK (ElasticSearch Logstash Kibana) [3]. Dicha solución es open source, lo que nos permite seguir cumpliendo el objetivo de utilización de una solución económica.

Tiempo de trabajo: 20/03/2020 al 21/03/2020

5. Documentación del proceso anterior.

Tiempo de trabajo: 22/03/2020 al 30/03/2020

- B. **Segunda fase.** Desarrollo y despliegue del software que enumerará los activos y servicios.

Tiempo de trabajo: del 01/04/2020 al 08/05/2020

Ésta segunda fase se compone de varios subfases:

1. Desarrollo del software de la aplicación de enumeración.

Tiempo de trabajo: 01/04/2020 al 30/04/2020

2. Instalación del código dentro de la sonda de enumeración.

Tiempo de trabajo: 01/05/2020 al 03/05/2020

Dentro de éste punto se explicará la instalación a realizar así como la generación de un demonio que permita lanzar el código tras un reinicio (controlado o no), permitiéndonos cumplir el objetivo de mantener el servicio activo 24*7*366

3. Documentación del proceso anterior.

Tiempo de trabajo: 04/05/2020 al 08/05/2020

- C. **Tercera fase.** Instalación de la máquina virtual que trabajará como: servidor de base de datos y aplicación web.

Tiempo de trabajo: del 09/05/2020 al 25/05/2020

Para tener este entorno perfectamente instalado se seguirán las siguientes premisas:

1. Se utilizará un sistema operativo Linux (servidor) sin interfaz gráfica, por su poca necesidad de recursos y su carácter gratuito.

NOTA: El sistema que se ha utilizado tiene 2Gb de RAM, 2 procesadores y el espacio de disco duro que define VMWare por defecto (20 Gb). Este valor siempre puede ser modificado al alza.

2. Se abrirá el puerto 22/TCP para la administración remota.

NOTA: Adicionalmente este puerto será bastionado

Tiempo de trabajo (2 puntos anteriores): 09/05/2020 al 12/05/2020

3. Se instalará la herramienta: **Elasticsearch**, que dentro de la solución ELK (Elasticsearch Logstash Kibana) [\[3\]](#), trabaja como base de datos.
4. Se instalará la herramienta: **Kibana** que dentro de la solución ELK (Elasticsearch Logstash Kibana) [\[3\]](#), permite tener un entorno web que nos servirá ver la información almacenada.

NOTA: Como se ha comentado anteriormente, las dos herramientas anteriores nos permite seguir cumpliendo el objetivo de utilización de una solución económica.

Tiempo de trabajo (2 puntos anteriores): 15/05/2020 al 18/05/2020

5. Se explicará la generación de un demonio que permita lanzar los anteriores servicios tras un reinicio controlado o no. Lo que nos permite cumplir el objetivo de mantener el servicio activo 24*7*366

Tiempo de trabajo: 13/05/2020 al 15/05/2020

6. Se instalará y configurará, como proxy-web, la herramienta: **ngix** [4], open source, que redirige las comunicaciones hechas al puerto 80/TCP del proxy-web hacia el puerto 5601/TCP, abierto por defecto por Kibana, y que nos permite acceder al entorno web manejado en la solución.

Tiempo de trabajo: 16/05/2020 al 17/05/2020

7. Documentación del proceso anterior.

Tiempo de trabajo: 18/05/2020 al 25/05/2020

Gráficamente, el trabajo a realizar sería [5]:

Nombre	Duración	Inicio	Terminado
Instalación de la sonda de enumeración	18,666 days	12/03/20 8:00	30/03/20 23:59
Instalación del sistema operativo y bastionado del servicio SSH	5,666 days	12/03/20 8:00	17/03/20 23:59
Instalación de la herramienta: nmap	1,666 days	18/03/20 8:00	19/03/20 23:59
Instalación de la herramienta: logstash	1,666 days	20/03/20 8:00	21/03/20 23:59
Generación de la documentación	8,666 days	22/03/20 8:00	30/03/20 23:59
Desarrollo y despliegue del software de enumeración	37,666 days	1/04/20 8:00	8/05/20 23:59
Desarrollo de nuevo código	29,666 days	1/04/20 8:00	30/04/20 23:59
Instalación dentro de la sonda y generación del demonio de control	2,666 days	1/05/20 8:00	3/05/20 23:59
Generación de la documentación	4,666 days	4/05/20 8:00	8/05/20 23:59
Instalación de la base de datos y del servidor web	16,666 days	9/05/20 8:00	25/05/20 23:59
Instalación del sistema operativo y bastionado del servicio SSH	3,666 days	9/05/20 8:00	12/05/20 23:59
Generación del demonio del control	2,666 days	13/05/20 7:00	15/05/20 22:59
Instalación y configuración del proxy-web	1,666 days	16/05/20 8:00	17/05/20 23:59
Generación de la documentación	7,666 days	18/05/20 8:00	25/05/20 23:59

Ilustración 1: Plan de trabajo - Tareas

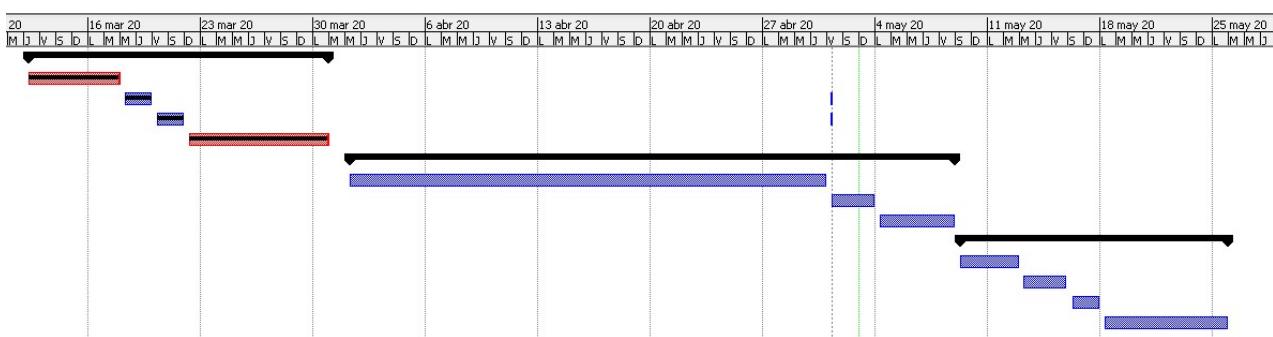


Ilustración 2: Plan de trabajo - Tiempos

1.7. Flujos de información del proyecto

Se procede a mostrar los flujos de las comunicaciones que se generaran en el presente proyecto.

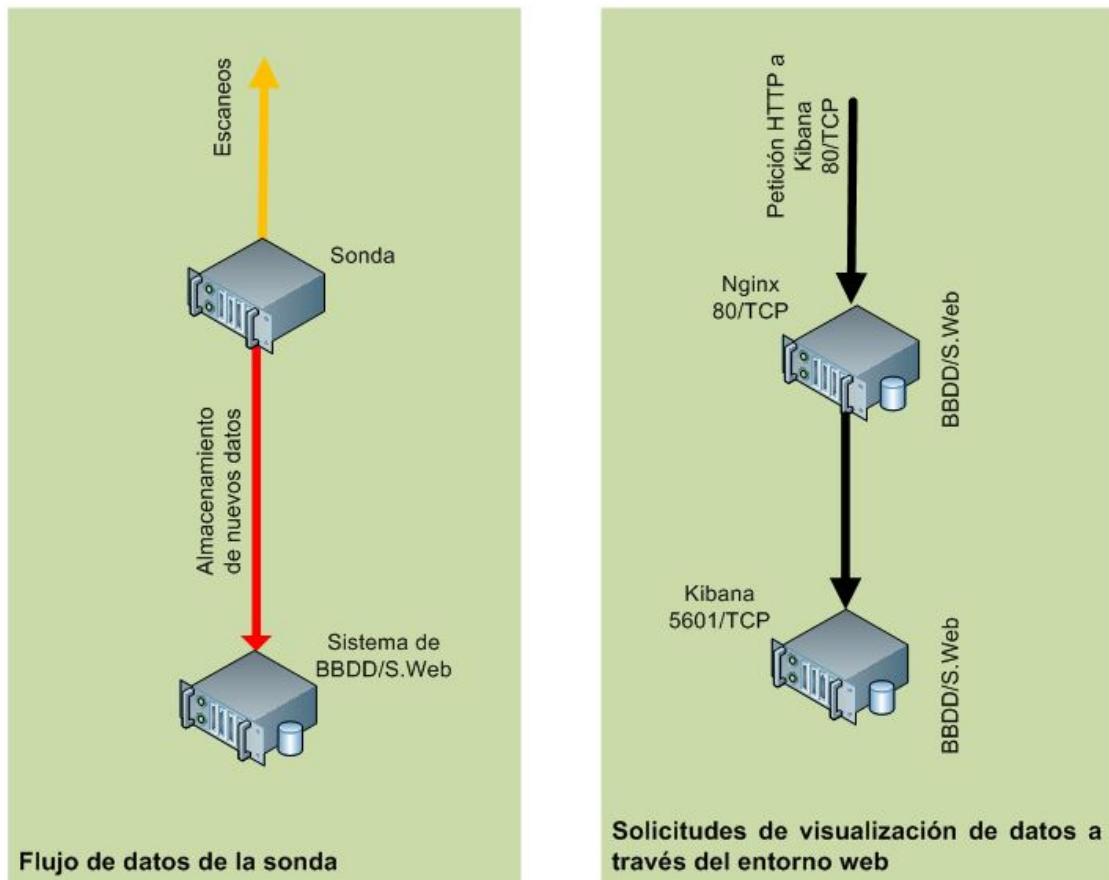


Ilustración 3: Flujos de comunicaciones del proyecto

2. FASE 1: SONDA DE ENUMERACIÓN

En esta fase se:

- Creará la máquina virtual que nos servirá para lanzar el proceso de enumeración/descubrimiento de activos. Esto se puede ver en el [Anexo I, parte 1](#)
- Se realizará la instalación del sistema operativo. Esto se puede ver en el [Anexo I, parte 2](#)
- Se instalará la herramienta: **nmap**. Esto se puede ver en el [Anexo I, parte 3](#)
- Se instalará y configurará la herramienta: **logstash**. Esto se puede ver en el [Anexo I, parte 4](#)
- Se bastionará de **servicio SSH**. Esto se puede ver en el [Anexo I, parte 5](#)
- Se creará un usuario específico para almacenar los distintos scripts que se van a crear y que se verá en el siguiente punto de este TFM.

El usuario que se creará tendrá el nombre: **Sonda**

Dentro de este punto cada remarcar la configuración de la herramienta: **logstash**, que permite enviar los datos recopilados a la base de datos ElasticSearch para su almacenamiento.

Como se comentará en la siguiente fase, la fase 2, la información obtenida se almacena en el archivo: **cit_resources.csv**. Como se puede suponer, dicho archivo tiene los datos recabados separados con “,”.

Una muestra de ello, sería:

	cit_resources.csv
1	2018-11-13 11:50:05,10.32.17.3,tcp,111,rpcbind,2-4
2	2018-11-13 11:50:05,10.32.17.3,tcp,139,netbios-ssn,Samba smbd
3	2018-11-13 11:50:05,10.32.17.3,tcp,13,daytime,Sun Solaris daytime
4	2018-11-13 11:50:05,10.32.17.3,tcp,199,smux,HP-UX smux
5	2018-11-13 11:50:05,10.32.17.3,tcp,21,ftp,Sun SunOS ftpd,5.6
6	2018-11-13 11:50:05,10.32.17.3,tcp,23,telnet,Sun Solaris telnetd
7	2018-11-13 11:50:05,10.32.17.3,tcp,37,time

Ilustración 4: Parte de información recaba

Debido al uso de este archivo para la captación de la información, la configuración [14] que debe establecerse para la herramienta: **logstash**, debe ser la siguiente.

```
GNU nano 2.9.3                                     csv.config

input {
    file {
        path => "/home/Sonda/cit_resources.csv"
    }
}

filter {
    csv {
        columns => [ "Fecha", "IP", "Protocolo", "Puerto", "Servicio", "Producto", "Version$"
    }
}

output {
    elasticsearch {
        hosts => ["10.39.101.50:9200"]
        index => "poc1"
    }
}
```

Ilustración 5: Configuración de Logstash

En la configuración, la sección: **input**, determina el origen de la información a enviar.

En nuestro caso, se corresponde con el archivo comentado anteriormente.

El apartado: **filter**, indica la categorización de los datos que van a ser enviados, el nombre que establezcamos aquí, será mostrado en Kibana como cabecera de las columnas.

El apartado: **output**, permite configurar la ubicación de la base de datos dónde se almacenará la información recabada. A la información enviada, a cada línea de información recabada, se le añade la cadena: “Sonda1”.

Es nuestro caso, no aporta mucho, pero si tuviéramos varias sondas desplegadas, podría servirnos para determinar el origen de cada información recibida.

Tras la instalación del servidor de ésta solución expuesta aquí, tendremos que volver a nuestra sonda para comprobar que las comunicaciones entre dicha sonda y la base de datos se realizan satisfactoriamente.

Resulta interesante mostrar como probar las conexiones contra la base de datos mediante la ejecución del siguiente comando:

```
./logstash -e "input { stdin { } } output { elasticsearch { hosts =>
["10.39.101.50:9200"] } }"
```

```
root@mishodan-sonda:/home/Sonda# /usr/share/logstash/bin/logstash -e 'input { stdin {} } output { elasticsearch { hosts => ["10.39.101.50:9200"] }}'
```

Ilustración 6: Comprobación básica de las comunicaciones Logstash

Permitirá enviar cualquier cadena de texto hacia la base de datos, como por ejemplo: "Esto es una prueba"

	@timestamp per hour
Time ▾	_source
> May 20, 2020 @ 17:20:40.839	@version: 1 message: Esto sigue siendo una prueba @timestamp: May 20, 2020 @ 17:20:40.839 host: mishodan-sonda _id: 3uusMnIB8B_RDcNwJk5K _type: _doc _index: logstash-2020.05.20-000001 _score: -
> May 20, 2020 @ 17:18:41.830	@version: 1 message: Hola esto es una prueba @timestamp: May 20, 2020 @ 17:18:41.830 host: mishodan-sonda _id: nuuqMnIB8B_RDcNwWU7J _type: _doc _index: logstash-2020.05.20-000001 _score: -

Ilustración 7: Recepción de las comunicaciones básicas de Logstash

Tras comprobar el existo de la misma ya podríamos comprobar que la configuración establecida en el archivo: csv.conf, funciona adecuadamente.

Por lo que podremos lanzar el comando:

./logstash -f csv.conf

Y comprobar los resultados obtenidos.

The screenshot shows the Kibana Discover interface with the URL 10.39.101.50/app/kibana#/discover?_g=(filters:(),refreshInterval(pause:0),...). The left sidebar contains various icons for search, filters, and visualization. The main area displays five log entries from May 20, 2020, at 17:02:24.640 to 17:02:24.645. Each entry includes fields like IP, Protocolo, path, Fecha, Servicio, Puerto, host, message, _id, _type, _doc, _index, and _score.

Timestamp	IP	Protocolo	path	Fecha	Servicio	Puerto	host	message	_id	_type	_doc	_index	_score
> May 20, 2020 @ 17:02:24.640	10.32.17.237	tcp	/home/Sonda/cit_resources.csv.1ejemplo	2020-04-06 05:20:33	McAfee ePolicy Orchestrator Agent Activity Log httpd	8081	mishodan-sonda	Este es una prueba	M0ubMnIB8B_RDcNwbU1E	log	1	poc1	-
> May 20, 2020 @ 17:02:24.646	10.32.17.237	tcp	/home/Sonda/cit_resources.csv.1ejemplo	2020-04-06 12:40:05;10.39.97.57	Este es una prueba	printer	mishodan-sonda	12:40:05;10.39.97.57,tcp,515,printer,Este es una prueba	M-ubMnIB8B_RDcNwbU1S	log	1	poc1	-
> May 20, 2020 @ 17:02:24.645	10.32.17.237	tcp	/home/Sonda/cit_resources.csv.1ejemplo	2020-04-06 04:59:46	tcpwrapped	5060	mishodan-sonda	04:59:46,10.32.17.237,tcp,5060,tcpwrapped	MuubMnIB8B_RDcNwbU1S	log	1	poc1	-
> May 20, 2020 @ 17:02:24.644	10.32.17.237	tcp	/home/Sonda/cit_resources.csv.1ejemplo	2020-04-06 04:48:48	Microsoft Terminal Service	3389	mishodan-sonda	ms-wbt-server,Microsoft Terminal Service	L-ubMnIB8B_RDcNwbU1E	log	1	poc1	-
> May 20, 2020 @ 17:02:24.603	d		/home/Sonda/cit_resources.csv.1ejemplo	May 20, 2020 @ 17:02:24.603			mishodan-sonda	host: mishodan-sonda	MeubMnIB8B_RDcNwbU1S	log	1	poc1	-

Ilustración 8: Comprobación recepción del contenido del archivo "cit_resources.csv"

3. FASE 2: DESARROLLO DE LA APLICACIÓN

Nuestra solución global se aprovecha de varias soluciones open source como puede ser: **nmap** (herramienta para el escaneo de servicios) y **ELK** (entorno de almacenamiento y visualización de información), para formar otra solución que permita cumplir los objetivos marcados inicialmente.

Pero algo que necesita esta solución y que no se cubre con las herramientas seleccionadas son:

1.- A la hora escanear posibles activos, **se necesita ser lo más silencioso posible para evitar el bloqueo de nuestras comunicaciones**, por parte de las posibles soluciones de seguridad existente en la/s red/es donde se despliegue la sonda.

Para ello, no se puede utilizar directamente la herramienta: nmap.

Concretamente, no se puede utilizar configuraciones que impliquen el escaneo de todos los puertos de un activo directamente (uso del parámetro: -p-) ni configuraciones que impliquen el escaneo de todo un rango de red (uso de la nomenclatura CIDR [\[6\]](#)) **por varios inconvenientes**:

- A. Es altamente conocida, por lo que **existen muchos IOC para detectar su uso**
- B. Es verdad que es posible configurar la velocidad de escaneo de dicha herramienta, pero si utilizamos su versión más rápida, estamos generando demasiado ruido (enorme flujo de paquetes por la red), mientras que si utilizamos su versión más lenta, el tiempo requerido para realizar un escaneo se hace demasiado alto.
- C. Las velocidades intermedias podrían ser suficientemente rápidas, pero nos podemos encontrar con problemas de saturación del sistema operativo, lo que implica una degradación del sistema.

2.- **La información obtenida por: nmap, no se encuentra parseada.** Por lo que es necesario parsear la información obtenida para su posterior envío a la solución ELK. Concretamente, **se enviará a: elasticsearch, a través de logstash**.

Para solventar los anteriores inconvenientes, se desarrolla una solución propia compuesta de los siguientes archivos.

3.1. Shodan.sh

Este archivo contendrá el código expuesto completamente en el Anexo II, punto [1](#)

NOTA: El desarrollo se hace en base al lenguaje bash de Linux, un lenguaje sencillo y predeterminado dentro del sistema operativo Linux. Por lo que no tenemos coste alguno ni necesidad de instalar ningún software adicional.

Explicación del código

0.- Se define las variables que vamos a utilizar

```
#!/bin/bash

#Variable utilizada mientras se espera que reduzca el número de hilos lanzados de Nmap
durmiendo=0
#Variable que controla el número de procesos en paralelo, esto depende de la memoria del sistema
hilos="15"

rangos=(10)
```

Ilustración 9: Shodan.sh - Variables

NOTA: La variable: **Rangos**, es una lista, por lo que podemos llegar a introducir múltiples valores. Dicha variable establece el primer octeto de nuestras direcciones IP.

1.- Ante posibles fallos que provoquen el reinicio de la sonda, y con objeto de no estar iniciado constantemente y de manera manual el escaneo desde 0, se procede a trabajar con el fichero: **ControlIP.txt**, que guardará la última IP analizada.

Por lo que si el archivo existe, se recogerán los datos referentes a la última IP y se procederá a seguir con el ciclo.

```
if [ -f /home/Sonda/ControlIP.txt ]; then
    aa=$(cat /home/Sonda/ControlIP.txt | cut -d "." -f 1)
    bb=$(cat /home/Sonda/ControlIP.txt | cut -d "." -f 2)
    cc=$(cat /home/Sonda/ControlIP.txt | cut -d "." -f 3)
    dd=$(cat /home/Sonda/ControlIP.txt | cut -d "." -f 4)
```

Ilustración 5: Shodan.sh – Recogemos la última IP analizada

2.- Se procede a comprobar que los valores recogidos anteriormente se encuentre entre los valores: 0-255, que definen los posibles valores de una dirección IP.

```
if [ $bb = "" ] || [ $cc = "" ] || [ $dd = "" ] || [ $bb -lt 0 ] || [ $bb -gt 254 ] || [ $cc -lt 0 ] || [ $cc -gt 254 ] || [ $dd -lt 0 ] || [ $dd -gt 254 ]; then
    echo "Hay un problema con alguno de los valores ubicado en el fichero: ControlIP.txt" > ./Logs/Logs
else
```

Ilustración 10: Shodan.sh – Control del formato de la última IP analizada



Añadir que, como se puede ver en el código, **se maneja un directorio de Logs, donde se almacenarán los fallos que pudiéramos tener en el momento de la ejecución del script.**

3.- Tras extraer la dirección IP almacenada en el archivo “ControlIP.txt”, se procede a ejecutar para cada puerto existente de cada IP del rango /8 definido a través de la variable “rangos”, un escaneo mediante “nmap”, con la intención de determinar si dicho puerto/servicio se encuentra abierto.

```
else
    while true; do
        for b in `seq $bb 254`; do
            for c in `seq $cc 254`; do
                for d in `seq $dd 254`; do
                    for port in `seq 1 65535`; do

                        if [ $(ps -ax | grep "nmap" | wc -l) -lt $hilos ]; then
                            echo ${rangos[0]}.$b.$c.$d":"$port > /home/Sonda/Logs/LogsEscaneos.logs
                            $(nmap -Pn -sV -open -p $port ${rangos[0]}.$b.$c.$d -oX
                            /home/Sonda/Resultados/${rangos[0]}.$b.$c.$d-$port.nmap 1> /dev/null 2> /dev/null &)
                        else
                            while [ $(ps -ax | grep "nmap" | wc -l) -ge $hilos ]; do
                                durmiente=$((durmiente + 1))
                            done
                            fi
                        done
                    echo ${rangos[0]}."$b"."$c"."$d" > /home/Sonda/ControlIP.txt
                    done
                    dd=0
                    done
                cc=0
                done
            bb=0
            done
        fi
    else
        echo "No existe el fichero que permite el comienzo del script" > /home/Sonda/Logs/Logs-AccesosFicheros
    fi
```

Ilustración 11: Shodan.sh – Enumeración de todas las IPs de un rango

Por la problemática con el software “nmap” descrito anteriormente en este mismo apartado, se desarrolla una gestión de procesos controlada mediante un desarrollo propio. Este desarrollo, principalmente consiste en lanzar en paralelo varios procesos de la herramienta “nmap”, cada uno de los cuales consiste en escanear un puerto/servicio de un posible activo.

Con ello, además, conseguimos utilizar múltiples puertos origen de nuestra sonda, con lo que los posibles IOCs (Indicadores de compromiso/Indications of compromiso) cuyos parámetros detectan múltiples intentos de conexión desde una misma IP y desde un mismo puerto, no saltan (es uno de los problemas de la herramienta: nmap).

Para esto se ha definido la variable: **hilos**, que contiene el valor 15, y que significa que se lanzarán 14 procesos “nmap” en paralelo.

NOTA: Se ha comprobado que estos procesos más el resto del sistema operativo consumen aproximadamente 600KB de un total de un 1024 KB de RAM. Se podría ajustar un poco más la cantidad de procesos a lanzar y así aprovechar el resto de memoria RAM que nos queda.

```
top - 10:43:16 up 13 days, 23:41, 1 user, load average: 8,16, 8,71, 8,95
Tasks: 176 total, 10 running, 95 sleeping, 0 stopped, 0 zombie
%Cpu(s): 75,5 us, 24,4 sy, 0,0 ni, 0,0 id, 0,0 wa, 0,0 hi, 0,2 si, 0,0 st
KiB Mem : 1009396 total, 272888 free, 618568 used, 117940 buff/cache
KiB Swap: 2018300 total, 1588732 free, 429568 used. 258484 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
27940	root	20	0	77488	33340	8212	S	13,6	3,3	0:00.41	nmap
27946	root	20	0	77508	33388	8188	S	13,6	3,3	0:00.41	nmap
27959	root	20	0	77592	33336	8096	S	13,6	3,3	0:00.41	nmap
27953	root	20	0	77640	33404	8196	S	13,3	3,3	0:00.40	nmap
27934	root	20	0	77592	33160	7968	S	12,0	3,3	0:00.40	nmap
27965	root	20	0	75428	32604	7700	R	11,0	3,2	0:00.33	nmap
27921	root	20	0	77456	33404	8260	S	9,3	3,3	0:00.41	nmap
27915	root	20	0	78328	36652	8684	R	8,0	3,6	0:00.46	nmap
27971	root	20	0	70588	27780	7700	R	8,0	2,8	0:00.24	nmap
962	root	20	0	667500	253248	2424	S	6,3	25,1	860:46.06	ShodanCNP.sh
27977	root	20	0	62520	19784	7720	R	4,3	2,0	0:00.13	nmap
27983	root	20	0	53544	8732	5508	R	1,0	0,9	0:00.03	nmap
27984	root	20	0	667500	252424	1588	R	1,0	25,0	0:00.03	ShodanCNP.sh
8	root	20	0	0	0	0	R	0,3	0,0	38:28.82	rcu_sched
817	systemd+	20	0	71140	3164	2960	S	0,3	0,3	35:00.93	systemd-resolve

Ilustración 12: Shodan.sh - Uso de la memoria

Para controlar que sólo se ejecutan los 15 procesos, se contabiliza la cantidad de procesos de “nmap” en ejecución, mediante el comando: **ps –aux | grep “nmap” | wc -l**. Si es inferior a 15 se permite la ejecución de otro proceso “nmap”, si es mayor o igual, el script se mantiene a la espera hasta que existan en memoria menos procesos “nmap”.

```
for port in `seq 1 65535`; do
    if [ $(ps -ax | grep "nmap" | wc -l) -lt $hilos ]; then
        echo ${rango[0]}.${b}.${c}.${d}"$port > /home/Sonda/Logs/LogsEscaneos.logs
        $(nmap -Pn -sV -open -p $port ${rango[0]}.${b}.${c}.${d} -oX
        /home/Sonda/Resultados/${rango[0]}.${b}.${c}.${d}-$port.nmap 1> /dev/null 2> /dev/null &
    else
        while [ $(ps -ax | grep "nmap" | wc -l) -ge $hilos ]; do
            durmiente=$((durmiente + 1))
        done
    fi
done
```

Ilustración 13: Shodan.sh - Control de procesos de nmap

Añadir que, como se puede ver en el código, se maneja un directorio de Logs, donde se almacenará el archivo: **LogsEscaneads.logs**, que contendrá la última IP y puerto escaneado.

Así mismo, los resultados que obtengamos se almacenarán cada uno en un archivo cuyo nombre estará compuesto por: <IP><Puerto>.nmap. El contenido de estos archivos será parseado y enviado a posteriormente mediante el software “logstash” a la base de datos “elasticsearch” (residente en el servidor de base de datos y web) mediante el siguiente script.



3.2. ParseoShodan.sh

Este archivo contendrá el código expuesto completamente en el Anexo II, punto 2

NOTA: El desarrollo se hace en base al lenguaje bash de Linux, un lenguaje sencillo y predeterminado dentro del sistema operativo Linux. Por lo que no tenemos coste alguno ni necesidad de instalar ningún elemento/software adicional.

Explicación del código

0.- Se define las variables que vamos a utilizar.

En este caso las variables almacenaran la fecha y hora del sistema en la que se parsea la información y que serán añadidas a la base de datos

```
fecha=$(date +%Y-%m-%d)  
hora=$(date +%H:%M:%S)
```

Ilustración 14: ParseoShodan.sh - Variables

1.- **El resto de código permite coger los archivos con extensión “.nmap” almacenamos en el directorio: Resultados, y como se comentó en la explicación del archivo “MiShodan.sh”, almacenan los resultados de cada escaneo realizado.**

Recogeremos específicamente los siguientes valores:

- Fecha y hora (los valores de las variables declaradas anteriormente)
- IP
- Protocolo
- Puerto
- Servicio
- Producto + versión

Pero antes de enviarla a la base de datos, la almacenaremos en un archivo “temporal” de nombre: cit_resources.csv (con formato CSV, es decir, separados por coma [,]). Este archivo será el que posteriormente configuraremos dentro del archivo de configuración, valga la redundancia, del aplicativo: logstash, como archivo de origen de los datos a transferir.

También se puede comprobar que registramos en el archivo: LogParseo.logs, el último archivo parseado.

Y por último, pero no menos importante, eliminamos el archivo procesado.



```

for archivo in $(ls /home/Sonda/Resultados/*.nmap)
do
    ip=$(echo $archivo | cut -d "-" -f 1 | cut -d "/" -f 5)

    #Información almacenada a modo de Log
    echo $fecha;"$archivo":"$ip" > /home/Sonda/Logs/LogsParseo.logs

    #Se extrae la información que se almacenará en la base de datos.
    grep "state=open" $archivo | awk -F "protocol=" '{print $2}' | awk -F "portid="
    '{print $1" "$2}'# | cut -d ">" -f 1,3 | awk -F "><service name=" '{print $1" "$2}' |
    awk -F "product=" '{print $1" "$2}' | awk -F "version=" '{print $1" "$2}' | cut -d "="
    -f 1 | sed 's//,/g' | cut -d "," -f 2,4,6,8,10 | sed "s/^/$fecha $hora,$ip,/" >> /home
    /Sonda/cit_resources.csv

    rm $archivo
done

```

Ilustración 15: ParseoShodan.sh - Parseo de la información

2.- Por último, esperaremos 60 segundos para volver a lanzar la ejecución de este archivo. Con esto, haremos que el directorio “Resultados”, vuelva a contener información.

```
sleep 60
```

```
done
```

Ilustración 16: ParseoShodan.sh - Tiempo de espera

3.3. ControlEstado.sh

Este archivo contendrá el código expuesto completamente en el Anexo II, punto [3](#)

NOTA: El desarrollo se hace en base al lenguaje bash de Linux, un lenguaje sencillo y predeterminado dentro del sistema operativo Linux. Por lo que no tenemos coste alguno ni necesidad de instalar ningún elemento/software adicional.

Este código se desarrolla para controlar que el proceso generado por el script: "MiShodan.sh", no pare.

Para lo cual, lo primero es recopilar la hora y la fecha del sistema, así como el número de procesos "nmap", presentes en la memoria RAM

Si el valor es igual a 1, significaría que sólo está presente en la memoria el proceso: "**ps -aux | grep "nmap" | wc -l**", por lo que los procesos de escaneo, por cualquier motivo no se están lanzando.

Como la sonda tiene configurado un demonio (que se verá en la siguiente sección), entonces reiniciamos la sonda mediante el comando: **shutdown**.

Si por el contrario, continuamos detectando los procesos de escaneos, dejamos que este proceso hiberne durante 60 segundos.



3.4. Creación de un demonio propio

Como se ha comentado en el apartado anterior, se debe crear un demonio dentro del sistema para hacer que nuestra solución comience cada vez que el sistema se reinicie, ya sea por un fallo no controlado del mismo o bien porque nosotros hayamos solicitado su reinicio por el uso del script “ControlEstado.sh”.

Para esto hay que:

0.- Crear el script: **Lanzador-Shodan.sh**, dentro de la ubicación: **/etc/initd**, de la sonda. Esto debe hacerse con el usuario: **Root**.

```
#!/bin/bash

#Este programa es un mero lanzador de los dos componentes de los que se componente el desarrollo:
#MiShodan

#Lanzamos ShodanCNP.sh
nohup /home/Sonda/ShodanCNP.sh &
#Lanzamos ParseoShodan.sh
nohup /home/Sonda/ParseoShodan.sh &
#Lanzamos ControlEstado-ShodanCNP.sh
nohup /home/Sonda/ControlEstado-ShodanCNP.sh &
```

Ilustración 17: Lanzador-Shodan.sh

2.- Introducir el siguiente archivo: **rc.local**, dentro de la ubicación: **/etc**

```
#!/bin/bash

sh /etc/init.d/Lanzador-ShodanCNP.sh

exit 0
```

Ilustración 18: Rc.local

Las referencias utilizadas para conseguir realizar este punto han sido [\[7\]](#) y [\[8\]](#)

3.5. Creación de la carpeta: Sonda

Como medida de seguridad, se crea una carpeta especial para guardar nuestro proyecto como ya se ha podido ver en el código mostrado anteriormente, que es: **Sonda**.

Dicha carpeta, además, sólo puede ser accedida por el usuario: **root**, y sólo él tiene capacidad para trabajar con estos archivos.

```
root@mishodan-sonda:/home# mkdir Sonda
root@mishodan-sonda:/home# ls -l
total 8
drwxr-xr-x 5 soc  soc  4096 mar 29 11:53 soc
drwxr-xr-x 2 root root 4096 abr 27 10:18 Sonda
```

Ilustración 19: Creada carpeta: Sonda

```
root@mishodan-sonda:/home/Sonda# ls -l
total 60
-rw----- 1 root root 33941 abr  6 10:27 cit_resources.csv
-rwx----- 1 root root   606 feb  1  2019 ControlEstado-ShodanCNP.sh
-rw----- 1 root root    13 abr  8 05:40 ControlIP.txt
drwx----- 2 root root  4096 abr 27 10:25 Logs
-rwx----- 1 root root  1024 abr 27 10:21 ParseoShodanCNP.sh
drwx----- 2 root root  4096 abr 27 10:25 Resultados
-rwx----- 1 root root  2063 ene  8 07:28 ShodanCNP.sh
```

Ilustración 20: Contenido de la carpeta: Sonda

Si dicho directorio el código mostrado anteriormente no funcionaría correcta, y por lo tanto debería ser modificado para su correcto funcionamiento.

4. FASE 3: SERVIDOR WEB, BBDD Y PROXY-WEB

En esta fase se:

- Creará la máquina virtual que nos servirá para almacenar y presentar los datos recabados. Esto al ser igual que en la instalación de la sonda, se puede ver en el [Anexo I, parte 1](#)
- Se realizará la instalación del sistema operativo. Esto al ser igual que en la instalación de la sonda, se puede ver en el [Anexo I, parte 2](#)
- Se bastionará de **servicio SSH**. Esto se puede ver en el [Anexo I, parte 5](#)
- Se instalará y configurará el entorno **ElasticSearch**. Esto se puede ver en el [Anexo III, parte I](#)

En el archivo de configuración de este entorno es importante establecer el parámetro:

Discovery.type: single-node

```
#discovery.seed_hosts: ["host1", "host2"]
#
#discovery.type: single-node
#
# Bootstrap the cluster using an initial set of master-eligible nodes:
#
```

Nuestro entorno necesita que la información fluya desde la sonda al servidor, esto implica que la configuración del servicio “ElasticSearch” esté escuchando por cualquier interfaz del servidor. Para ello se configura el parámetro “Network.host” a 0.0.0.0 ó 0. Pero, dicha configuración provoca un error al iniciar el servicio, pero estableciendo el parámetro “Discovery.type” tal y como se ha mostrado evitamos esa problema. [\[9\]](#) [\[10\]](#) [\[11\]](#)

- Se instalará y configurará el entorno **Kibana**. Esto se puede ver en el [Anexo III, parte II](#)
- Se instalará y configurará el entorno **Nginx**. Esto se puede ver en el [Anexo III, parte III](#)

Para mejorar la seguridad del entorno del servidor se ha procedido a:

- En el entorno: Nginx

Este entorno nos permitirá bastionar un poco más nuestra solución, ya que cuando se realiza la instalación del Kibana, el acceso a dicho entorno se realiza sin la necesidad de credenciales.

El método para bastionar nuestro acceso a Kibana mediante el entorno Nginx, será mediante el uso de credenciales de autenticación. Para lo cual debemos configurar el entorno Nginx como proxy-web.

Para ello debemos cambiar la configuración por defecto.

El archivo de configuración reside en: /etc/nginx/sites-available, y posee el nombre: default.

La configuración utilizada para configurar Nginx como proxy-web, es:

```
GNU nano 2.9.3                               /etc/nginx/sites-available/default

# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration
#
server {
    listen 80 default_server;

    server_name Acceso_Servidor_Sonda;

    auth_basic "Acesso a Kibana";
    auth_basic_user_file /etc/nginx/htpasswd.kibana;

    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        #try_files $uri $uri/ =404;
        proxy_pass http://localhost:5601;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }

    # pass PHP scripts to FastCGI server
    #
    #location ~ \.php$ {
    #    include snippets/fastcgi-php.conf;
    #
    #    # With php-fpm (or other unix sockets):
    #}
```

Ilustración 21: Configuración utilizada en Nginx



De esta configuración los puntos más importantes son:

- 1.- **auth_basic_user_file <archivo>**: En el archivo aquí indicado residen los usuarios/contraseñas tienen permiso para dirigirse al servicio web de Kibana
- 2.- **proxy_pass <URL>**: Con esto estamos indicando que tras la validación satisfactoria, seremos redirigidos el puerto 5601/TCP, correspondiente al entorno web de Kibana

El archivo de donde reside el usuario y su contraseña, se puede crear mediante el comando:

```
htpasswd -c /etc/nginx/htpasswd.kibana adminkibana
```

Tras la configuración, al visitar el servicio establecido sobre el puerto 80/HTTP, podemos ver:

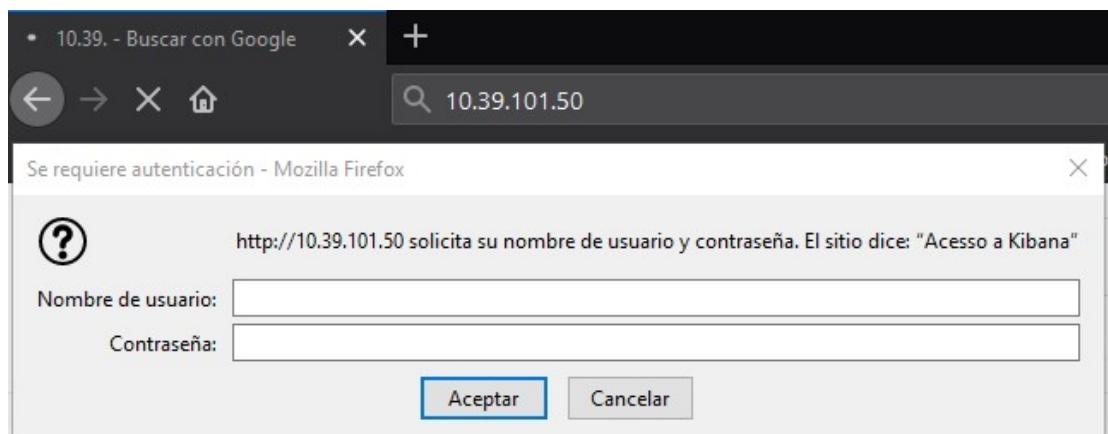


Ilustración 22: Proxy-web tras el puerto 80 del servidor del proyecto

- En el entorno: Kibana

Este entorno nos permitirá bastionar un poco más nuestra solución, ya que cuando se realiza la instalación del Kibana, podemos seleccionar si el servicio se verá de manera pública o de manera privada.

En nuestro caso, configuraremos el servicio de manera privada, es decir, haremos que el servicio sólo sea accesible desde el propio servidor montado.

Para ello es necesario establecer dentro del archivo de configuración del servicio el campo:

```
server.host: localhost
```

```
GNU nano 2.9.3                               /etc/kibana/kibana.yml

# Kibana is served by a back end server. This setting specifies the
server.port: 5601

# Specifies the address to which the Kibana server will bind. IP
# The default is 'localhost', which usually means remote machines
# To allow connections from remote users, set this parameter to a
server.host: "localhost"
```

Por lo tanto, sólo se podrá acceder al entorno de Kibana a través del servicio Nginx.

Añadir que **este entorno es donde trabajaremos en la catalogación de nuestros activos, descubriendo aquellos que son críticos de los que no.**

En cualquier auditoria, los entornos críticos son aquellos que están relacionados con servicios dados, por lo tanto podemos hacer una catalogación previa y **configurar un “dashboard” dentro de Kibana en donde aparezcan dichos servicios.**

En un primer momento, podemos configurar dicho dashboard para que nos muestre servicios como:

- Servidores FTP (puerto 21/TCP)
- Servidores SSH (puerto 22/TCP)
- Servidores Web basados en protocolo HTPP y HTTPS (puertos 80/TCP y 443/TCP)
- Servidores SMB (puertos 137/TCP, 139/TCP y 445/TCP)

5. FUTUROS PASOS

Los siguientes pasos de este trabajo, consistirán en:

- **Bastionar el acceso a Kibana** para poder manejar credenciales de acceso. Esto permitiría tener el segundo factor de autenticación dentro de nuestra solución.

Para ello podemos utilizar el plugin: x-pack, que en la última versión ya viene instalado.

```
root@mishodan-servidor:/usr/share/kibana/bin# ./kibana-plugin install --allow-root x-pack
Plugin installation was unsuccessful due to error "Kibana now contains X-Pack by default, there is no longer any need to install it as it is already present."
```

- **Bastionar el servidor web Kibana**, mediante el uso del protocolo HTTPS, por lo que necesitaríamos certificados, auto-firmados o no.

- **Bastionar el acceso al Elasticsearch** mediante:

1.- **Credenciales de acceso.**

2.- **Uso de comunicaciones seguras mediante el uso de cifrado.**

El uso de certificados sería tanto a nivel de servidor como a nivel de la sonda, ya que con ello la sonda certifica que el servidor es válido y viceversa.

- **Mejorar la clasificación de los activos** críticos de los que “no” lo son, mediante:

- La inclusión de la detección del sistema operativo dentro en las peticiones del script: Shodan.sh. Lo cual nos permitiría crear dashboard dentro de kibana específicos por tipos de sistemas operativos de servidores.
- La creación de distintos dashboards en función de los servicios encontrados.

- **Continuar con el ciclo de vida de la aplicación para su mejora continua.**

6. VALORACIONES PERSONALES

Este trabajo fin de máster me ha permitido:

- Ordenar las ideas para así poder presentarlas
- Llevar a la realidad una idea que surgió hace meses en mi cabeza, con lo que eso supone para mi trabajo actual o para mis trabajos futuros.
- Incrementar mis conocimientos sobre programación y sobre gestión de recursos del sistema operativo
- Aprender más sobre un entorno anteriormente desconocido para mí, el entorno ELK (Elasticsearch, Logstash y Kibana)
- Cumplir todos y cada uno de los objetivos que debía cumplir la solución y que fueron marcados inicialmente en este trabajo de fin de máster

ANEXO I: PROCESO DE INSTALACIÓN DE LA SONDA DE ENUMERACIÓN

AI.1. Creación de la máquina virtual

Se procede a crear la máquina virtual para el despliegue de la Sonda, para lo cual se tiene instalada la aplicación VMWare Workstation.

Tras validarse nos movemos a la opción: **File**, donde se seleccionará la opción: **New virtual machine**. (Mi versión del software se encuentra en inglés)



Ilustración 23: Creación MV - Paso 1

Dejaremos seleccionada la opción por defecto: **Typical (recommended)**, tras lo cual pulsaremos sobre el botón: **Next**.

En la ventana que nos aparece a continuación, seleccionaremos el sistema operativo a instalar, en nuestro caso será un sistema operativo: **Ubuntu Server 18.04**, descargado en imagen ISO

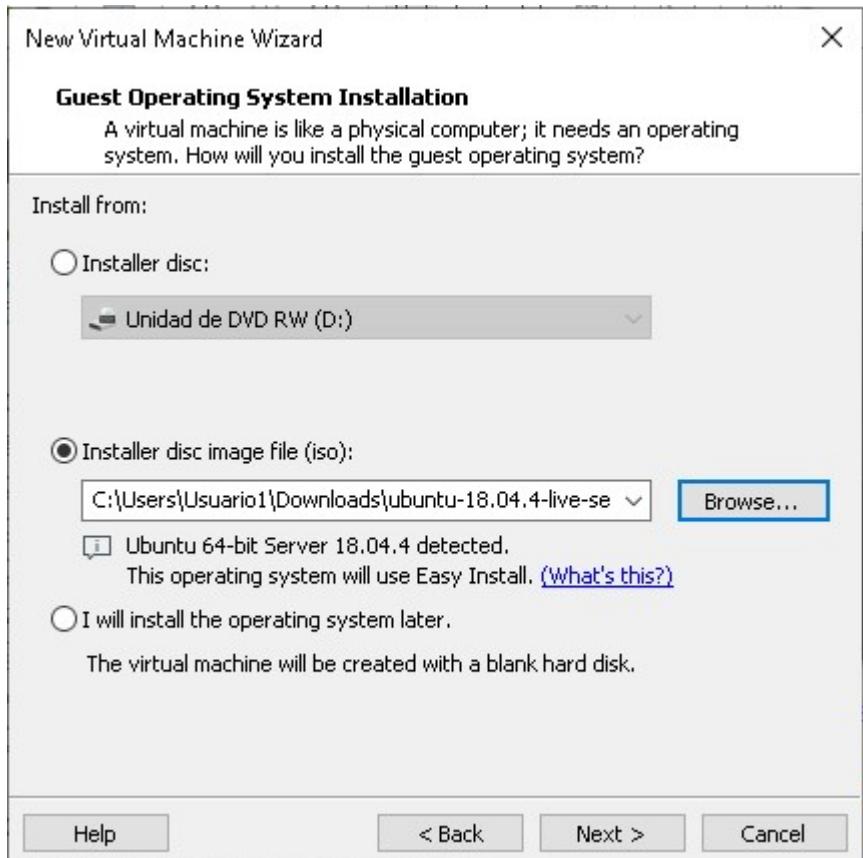


Ilustración 24: Creación MV - Paso 2

VMWare detectará que queremos instalar un sistema operativo Ubuntu y nos pedirá credenciales del usuario a crear en la nueva máquina. Aunque realmente no sería necesario porque al instalar el propio sistema operativo, éste nos lo volverá a solicitar.

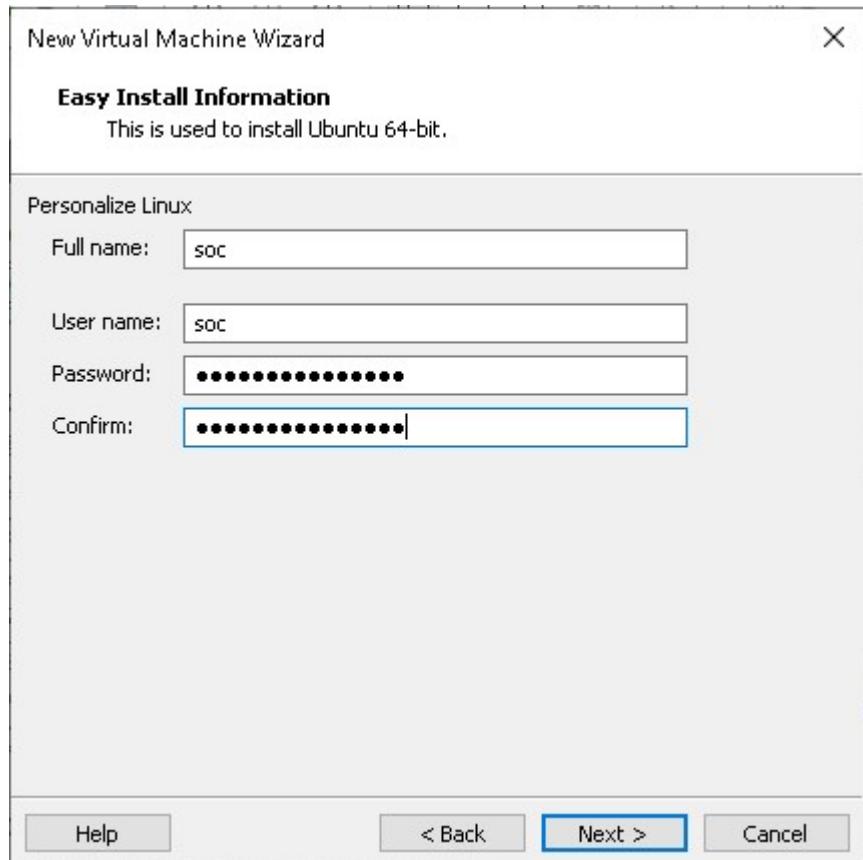


Ilustración 25: Creación MV - Paso 3

Tras pulsar sobre el botón: **Next**, deberemos introducir el nombre de la máquina virtual y su ubicación

Recomendación: Mi experiencia personal me ha mostrado que la mejor práctica es crear una partición independiente a la del sistema operativo del ordenador anfitrión, con capacidad para que nuestra máquina crezca. Por eso, se puede ver en la siguiente imagen el uso de la unidad G:

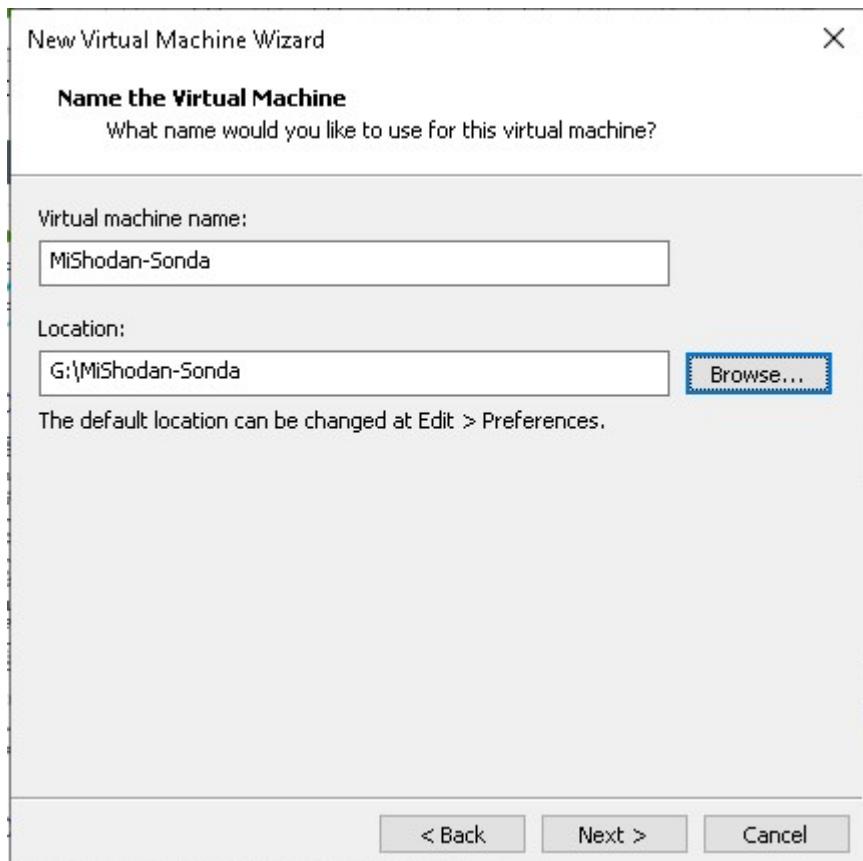


Ilustración 26: Creación MV - Paso 4

Tras pulsar sobre el botón: **Next**, se nos solicita definir el espacio de disco duro a reservar. Para las sondas, no se necesita mucho espacio, ya que el software desarrollado no requiera mucho espacio de disco duro, además de que la información recolectada será enviada a la base de datos y por lo tanto no será almacenada en local.

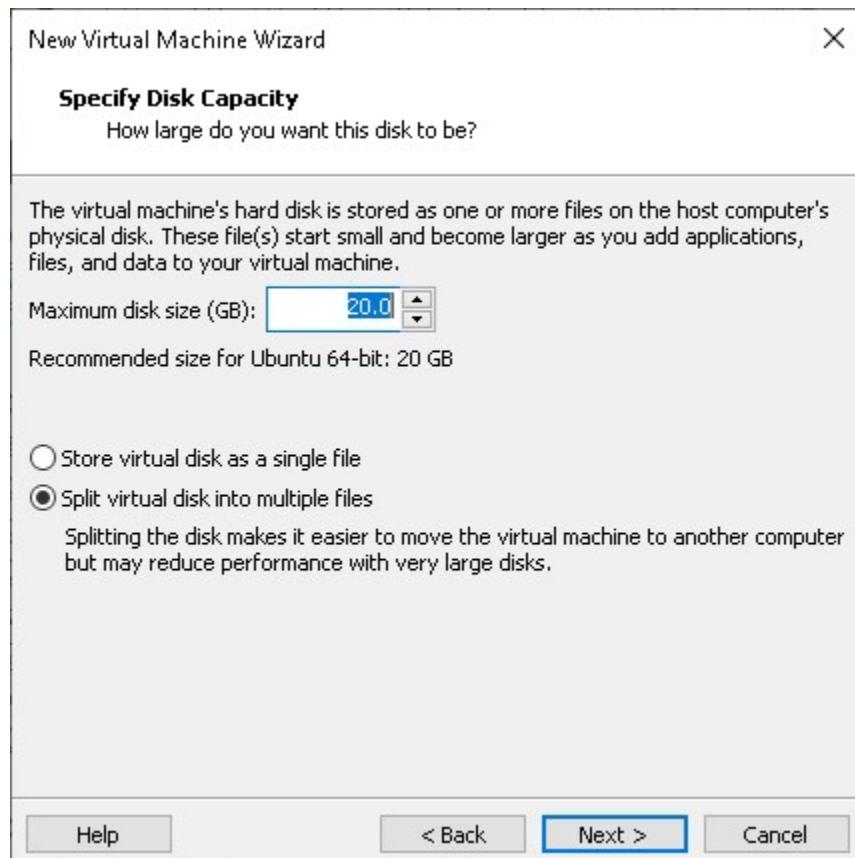


Ilustración 27: Creación MV - Paso 5

Tras pulsar sobre el botón: **Next**, se nos presentará un resumen de las características de nuestra nueva máquina virtual. Hay que decir que necesitaremos modificar las características de nuestra máquina, mediante la opción de: **Customize Hardware**, con la intención de otorgar a nuestra máquina virtual dos núcleos de CPU y una memoria RAM de 2048 bytes.

Estas dos opciones mejorarán el rendimiento de nuestra máquina virtual.

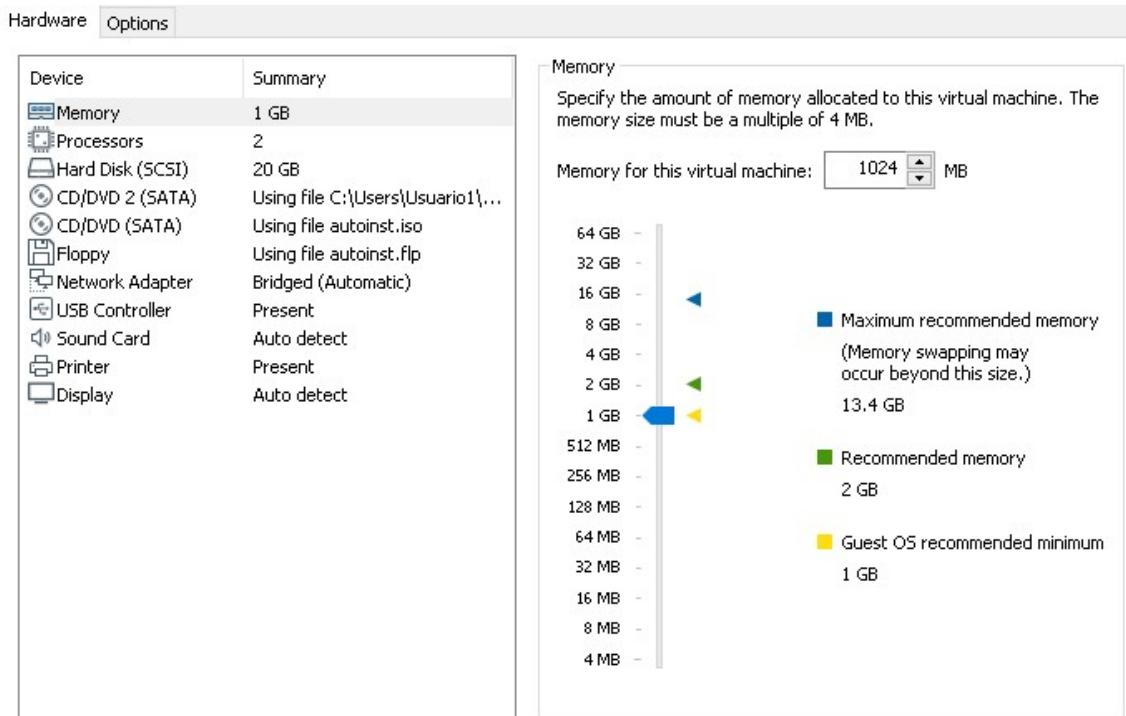


Ilustración 28: Creación MV - Paso 6

Tras pulsar sobre el botón: **Finalizar**, ya se tendrá creada la máquina virtual, y podremos empezar la instalación del sistema operativo.

AI.2. Instalación del sistema operativo

Tras el paso anterior, la primera pantalla de configuración se corresponde con el lenguaje a utilizar. En nuestro caso: **español**.

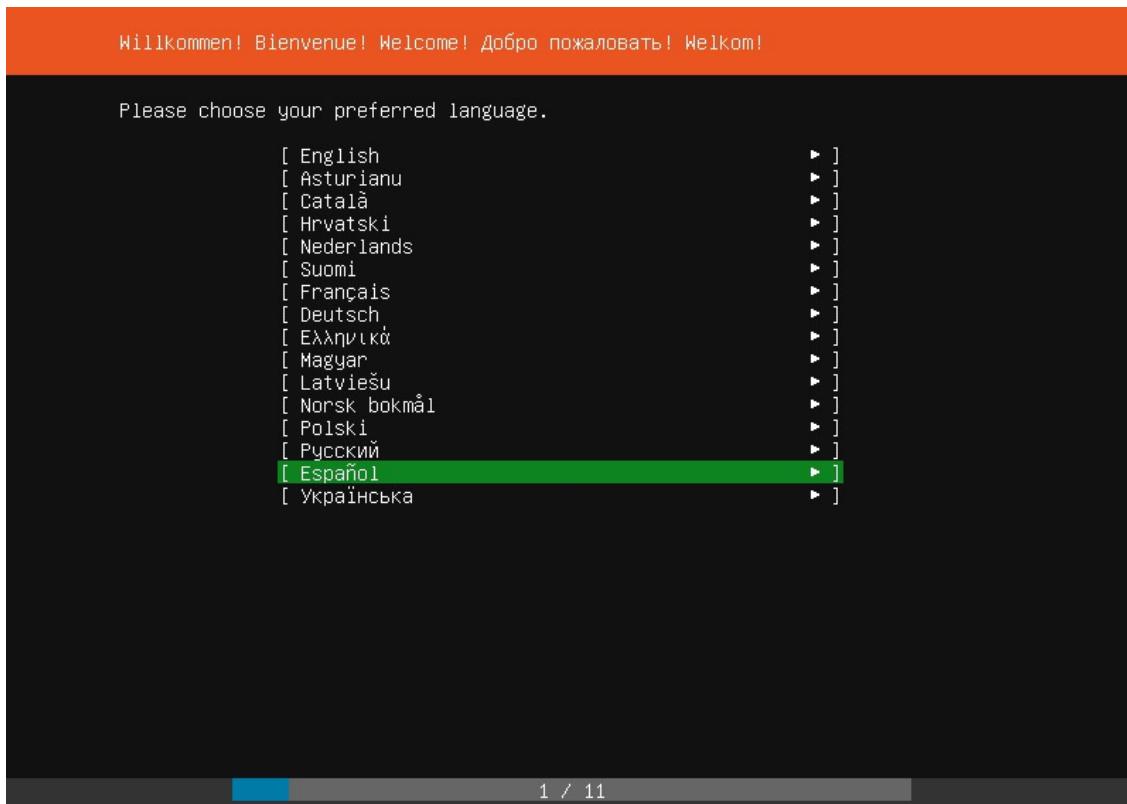


Ilustración 29: Instalación sistema operativo - Paso 1

La siguiente pantalla nos solicitará configurar el lenguaje del teclado, por supuesto; **español**

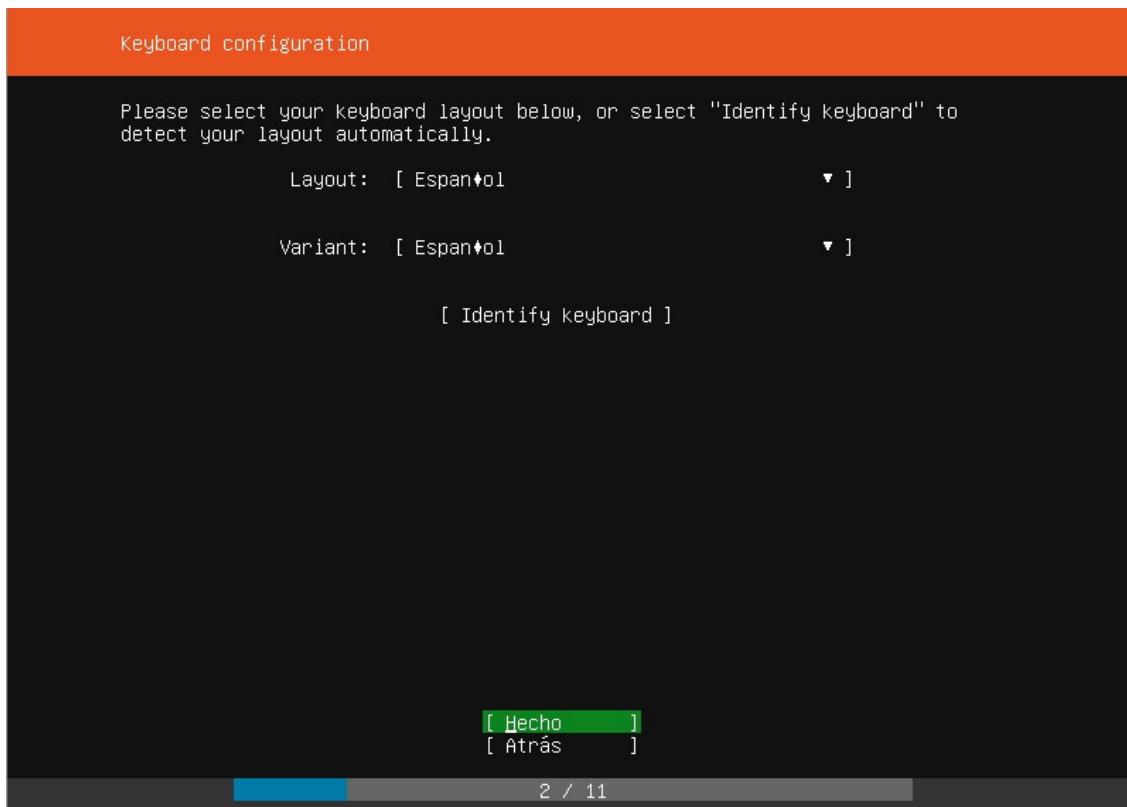


Ilustración 30: Instalación sistema operativo - Paso 2

La siguiente pantalla nos solicita el tipo de instalación queremos realizar, en nuestro caso: **Instalar Ubuntu**

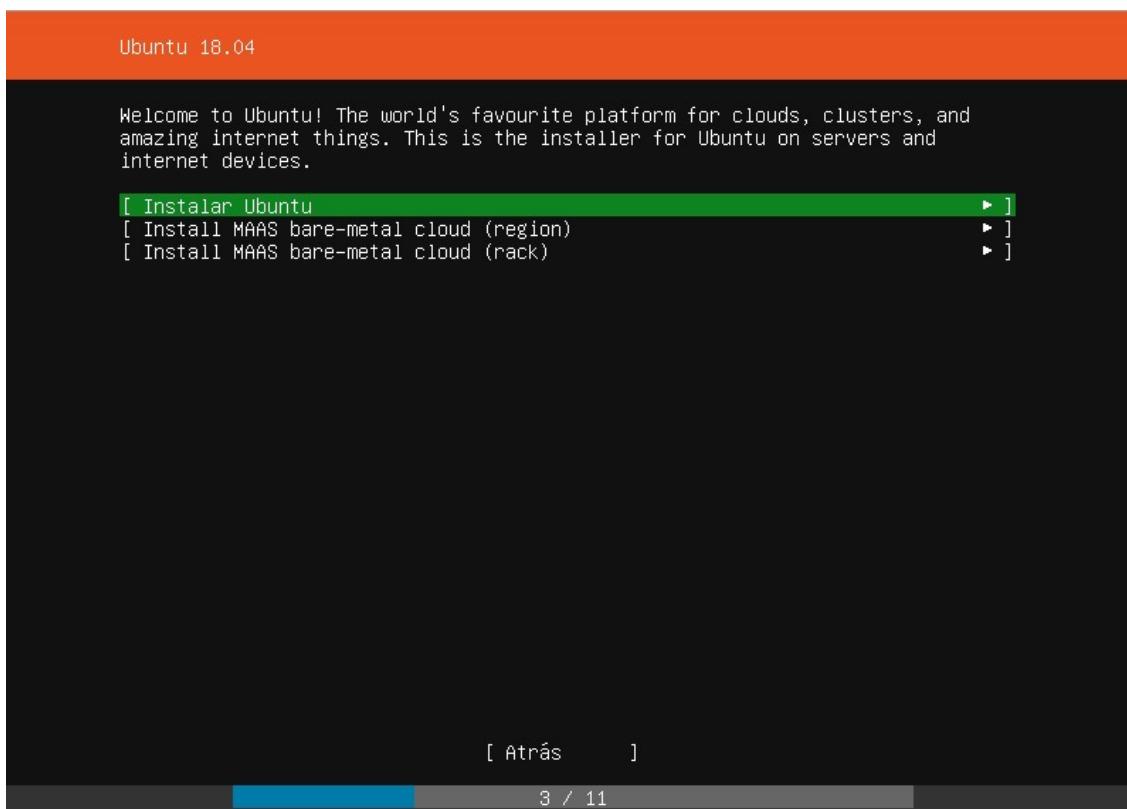


Ilustración 31: Instalación sistema operativo - Paso 3

La siguiente pantalla nos solicita configurar el interfaz de red que hemos indicado que existe al indicar las características de la máquina virtual.

La configuración será, sobre IPv4, la siguiente:

IP: 10.39.101.17

Máscara de red: 255.255.255.128

Puerta de enlace: 10.39.101.1

DNS: 10.200.10.3,10.200.10.4

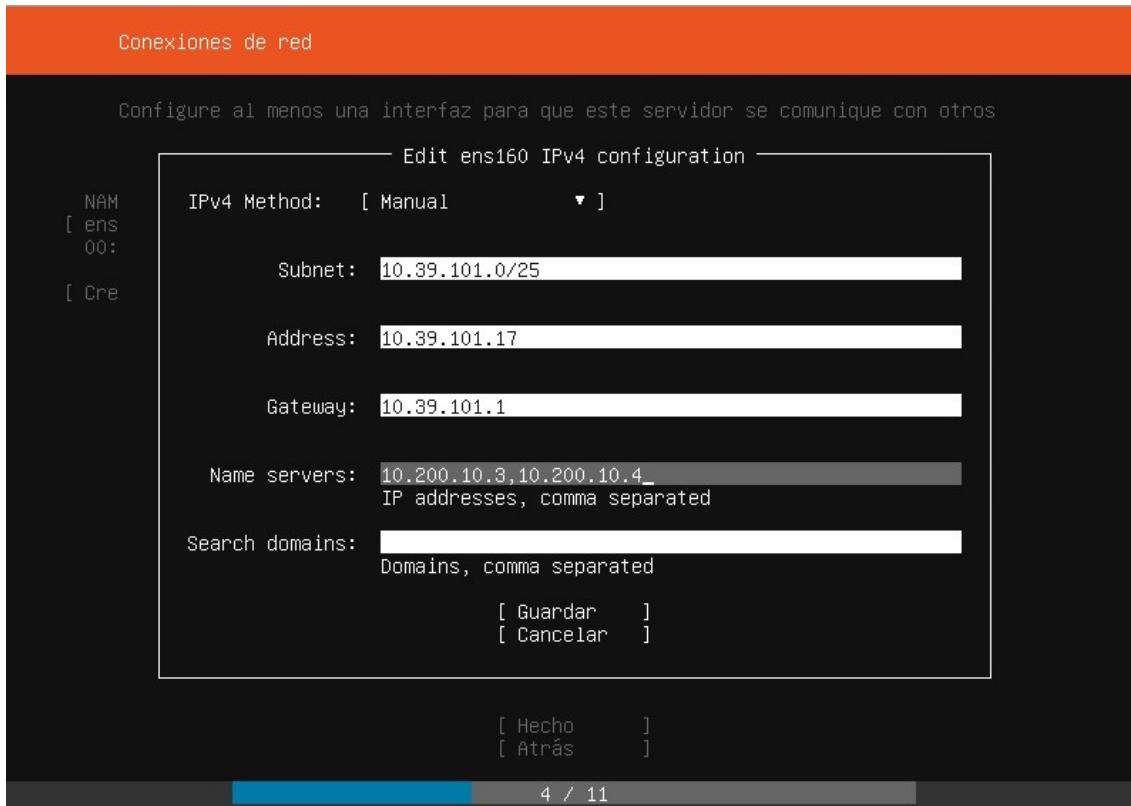


Ilustración 32: Instalación sistema operativo - Paso 4

La siguiente pantalla nos solicita configurar el proxy-web para la navegación hacia Internet, pero como no tenemos salida hacia Internet, lo dejamos en blanco.

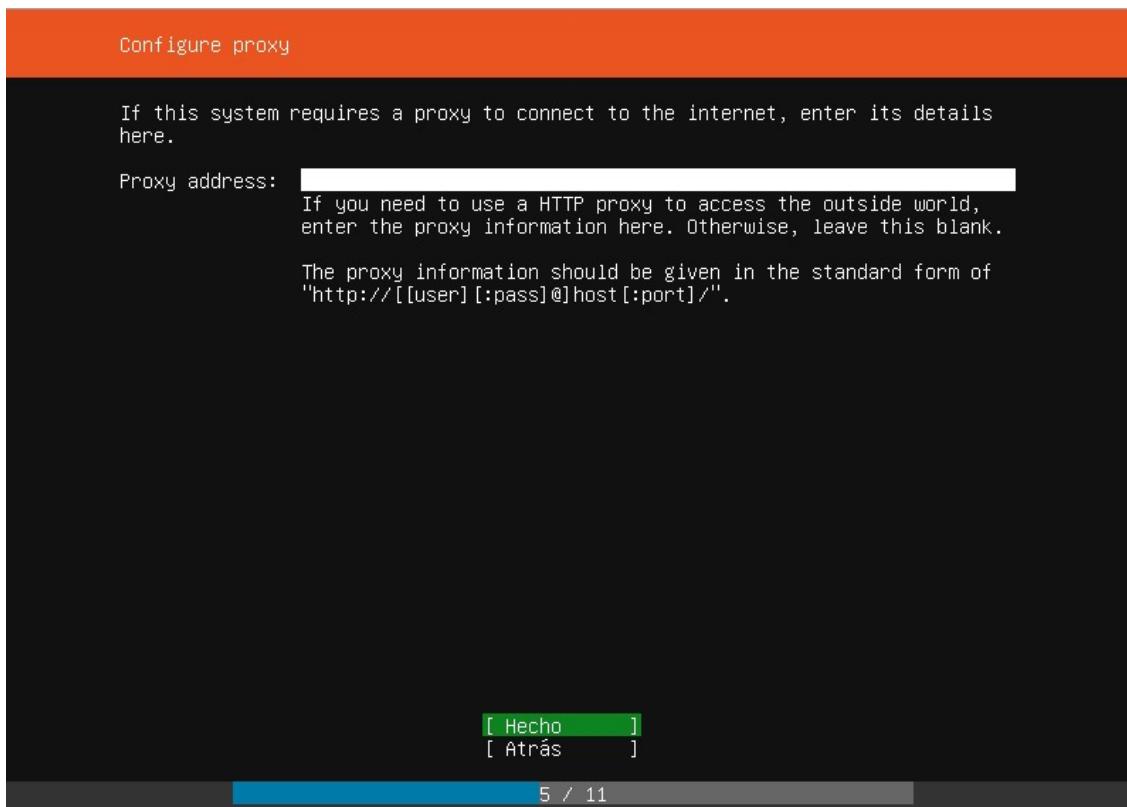


Ilustración 33: Instalación sistema operativo - Paso 5

La siguiente pantalla nos solicita introducir una ubicación alternativa que actué como: **sourcelist**, para las actualizaciones del sistema.

En nuestro caso, lo dejamos tal cual.

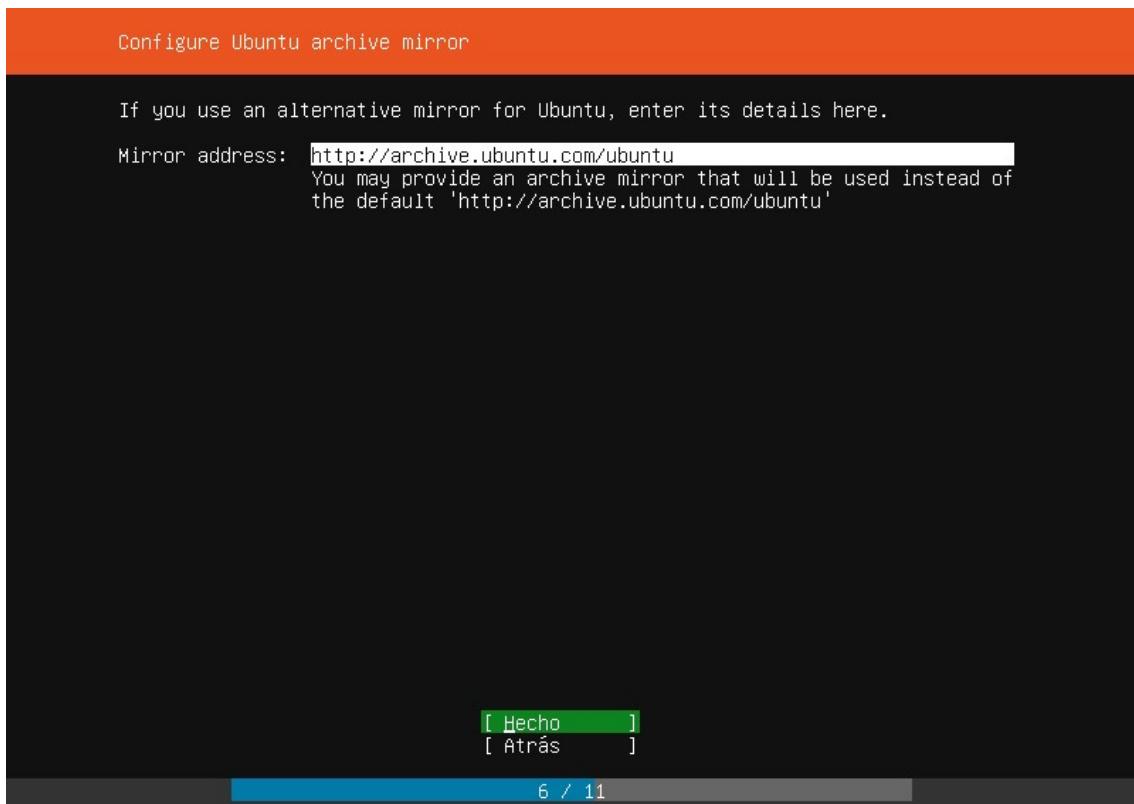


Ilustración 34: Instalación sistema operativo - Paso 6

La siguiente pantalla nos solicita configurar las particiones del disco a duro, se procede a dejar la configuración por defecto, es decir: **Use An Entire Disk**

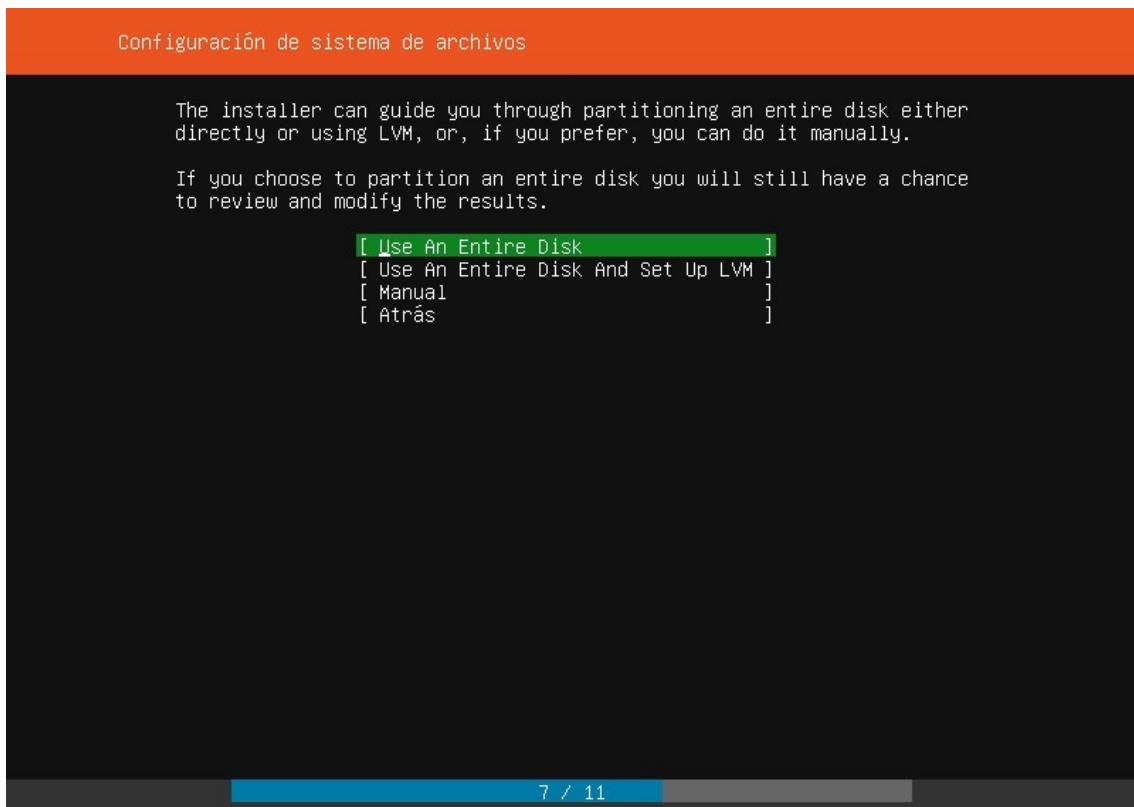


Ilustración 35: Instalación sistema operativo - Paso 7



Ilustración 36: Instalación sistema operativo - Paso 8

Configuración de sistema de archivos

RESUMEN DEL SISTEMA DE ARCHIVOS

MOUNT POINT	SIZE	TYPE	DEVICE	TYPE
[/]	19.997G	new ext4	new partition of local disk	►]

DISPOSITIVOS DISPONIBLES

No available devices

[Create software RAID (md) ►]
[Create volume group (LVM) ►]

USED DEVICES

DEVICE	TYPE	SIZE	►]
[/dev/sda]	local disk	20.000G	►]
partition 1 new, bios_grub		1.000M	►]
partition 2 new, to be formatted as ext4, mounted at /		19.997G	►]

Ilustración 37: Instalación sistema operativo - Paso 9

La siguiente pantalla nos solicita configurar el nombre del equipo, así como el usuario de uso habitual que podrá escalar privilegios a administrador

Los datos serán:

Nombre servidor: *mishodan-sonda*

Nombre usuario: *SOC*

Usuario: *soc*

Contraseña: *3ntr4alaSonda*

NOTA: Como se ha puede leer en la pantalla siguiente, este mismo usuario valdrá para conectarse remotamente a través del protocolo SSH

Configuración de perfil

Enter the username and password you will use to log in to the system. You can configure password is still needed for sudo.

Your name: soc

Your server's name: mishodan-sonda
The name it uses when it talks to other computers.

Pick a username: soc

Choose a password: *****

Confirm your password: *****

Ilustración 38: Instalación sistema operativo - Paso 10

SSH Setup

You can choose to install the OpenSSH server package to enable secure remote access to your server.

[X] Install OpenSSH server

Import SSH identity: [No]
You can import your SSH keys from Github or Launchpad.

Import Username:

[X] Allow password authentication over SSH

Ilustración 39: Instalación sistema operativo - Paso 11

La siguiente pantalla nos posibles paquetes adicionales a la instalación del sistema operativo, en este caso, no seleccionaremos nada.

Featured Server Snaps

These are popular snaps in server environments. Select or deselect with SPACE, press ENTER to see more publisher and versions available.

(_) microk8s	canonical✓	Kubernetes for workstations and appliances
() nextcloud	nextcloud✓	Nextcloud Server - A safe home for all your data
() wekan	xet7	Open-Source Kanban
() kata-containers	katacontainers✓	Lightweight virtual machines that seamlessly plug into
() docker	canonical✓	Docker container runtime
() canonical-livepatch	canonical✓	Canonical Livepatch Client
() rocketchat-server	rocketchat✓	Group chat server for 100s, installed in seconds.
() mosquitto	ralight	Eclipse Mosquitto MQTT broker
() etcd	canonical✓	Resilient key-value store by CoreOS
() powershell	microsoft-powershell✓	PowerShell for every system!
() stress-ng	cking-kernel-tools	A tool to load, stress test and benchmark a computer sys
() sabnzbd	sabnzbz	SABnzbd
() wormhole	snapcrafters	get things from one computer to another, safely
() aws-cli	aws✓	Universal Command Line Interface for Amazon Web Service
() google-cloud-sdk	google-cloud-sdk✓	Command-line interface for Google Cloud Platform produc
() slcli	softlayer	Python based SoftLayer API Tool.
() doctl	digitalocean✓	DigitalOcean command line tool
() conjure-up	canonical✓	Package runtime for conjure-up spells
() minidlna-escoand	escoand	server software with the aim of being fully compliant w
() postgresql10	cmd✓	PostgreSQL is a powerful, open source object-relational
() heroku	heroku✓	CLI client for Heroku
() keepalived	keepalived-project✓	High availability VRRP/BFD and load-balancing for Linux
() prometheus	canonical-is-snaps	The Prometheus monitoring system and time series databa
() juju	canonical✓	Simple, secure and stable devops. Juju keeps complexity

Ilustración 40: Instalación sistema operativo - Paso 12



La siguiente pantalla nos presentará un resumen del proceso de configuración del sistema operativo, y además, nos solicitará el reinicio del servidor.

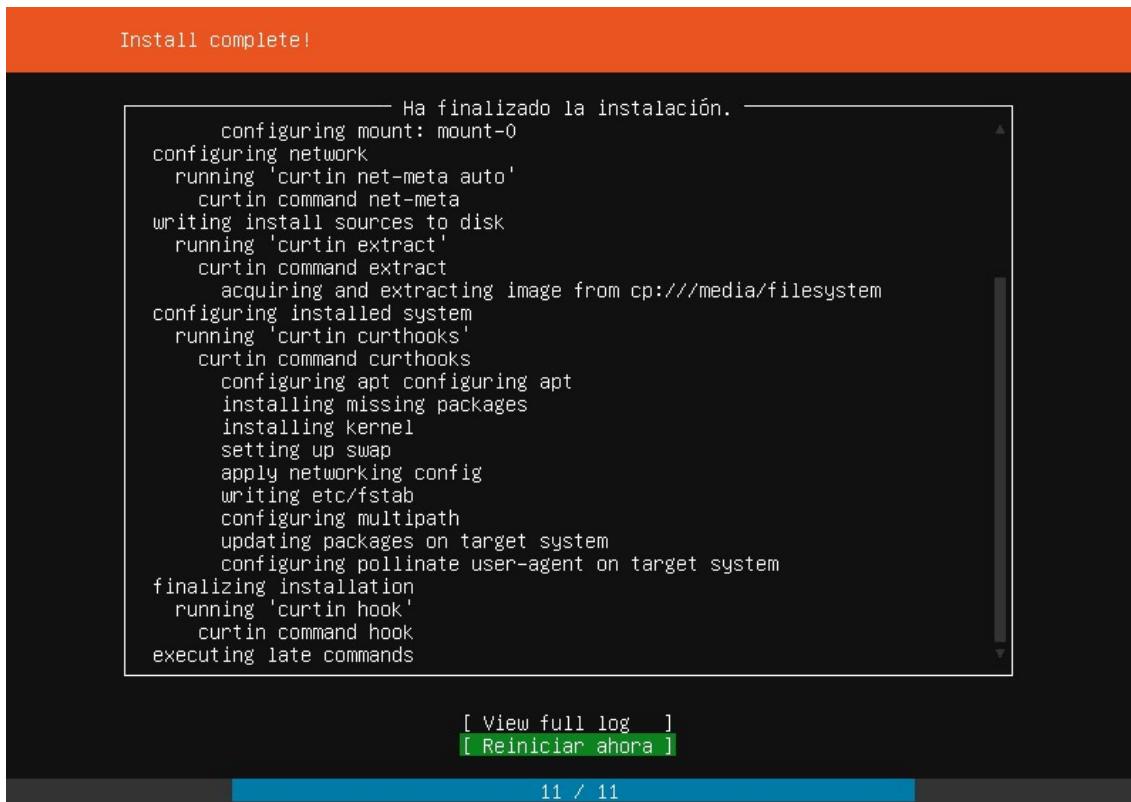


Ilustración 41: Instalación sistema operativo - Paso 13

Tras el reinicio de la máquina, se podrá ver la siguiente pantalla:

```
mishodan-sonda login: soc
Password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-88-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 System information as of Fri Mar  6 11:21:36 UTC 2020

 System load:  0.07          Processes:           193
 Usage of /:   19.8% of 19.56GB  Users logged in:    0
 Memory usage: 12%          IP address for ens33: 172.21.5.247
 Swap usage:   0%

14 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

soc@mishodan-sonda:~$
```

Ilustración 42: Instalación sistema operativo finalizada



AI.3. Instalación de la herramienta: NMAP

Para que el código desarrollado funcione correctamente es necesario instalar la herramienta: **nmap**. Para ello:

1.- Ejecutamos el comando: **sudo su**, para empezar a trabajar como el usuario **root** del sistema. Esto es necesario para poder instalar ciertas herramientas.

NOTA: el comando sudo, permite escalar privilegios, siempre y cuando el usuario con el que estemos lanzando dicho comando, este configurado dentro del archivo de configuración del comando: **sudo** [12].

```
soc@mishodan-sonda:~$ sudo su  
[sudo] password for soc:  
root@mishodan-sonda:/home/soc#
```

Ilustración 43: Nmap - Escalamos privilegios

2.- Ejecutamos el comando: **apt-get update**, que permitirá descargar las últimas actualizaciones de índices del sistema.

```
root@mishodan-sonda:/home/soc# apt-get update  
Obj:1 http://es.archive.ubuntu.com/ubuntu bionic InRelease  
Des:2 http://es.archive.ubuntu.com/ubuntu bionic-updates InRelease [88,7 kB]  
Des:3 http://es.archive.ubuntu.com/ubuntu bionic-backports InRelease [74,6 kB]  
Des:4 http://es.archive.ubuntu.com/ubuntu bionic-security InRelease [88,7 kB]  
Des:5 http://es.archive.ubuntu.com/ubuntu bionic/main Translation-es [364 kB]  
Des:6 http://es.archive.ubuntu.com/ubuntu bionic/restricted Translation-es [1.960 B]  
Des:7 http://es.archive.ubuntu.com/ubuntu bionic/universe Translation-es [1.259 kB]  
Des:8 http://es.archive.ubuntu.com/ubuntu bionic/multiverse Translation-es [74,9 kB]  
Descargados 1.952 kB en 2 s (1.283 kB/s)  
Leyendo lista de paquetes... Hecho  
root@mishodan-sonda:/home/soc#
```

Ilustración 44: Nmap - Actualizamos repositorios

3.- Ejecutamos el comando: **apt-get install nmap**, que permitirá la instalación de dicha herramienta.

```
root@mishodan-sonda:/home/soc# apt-get install nmap  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Se instalarán los siguientes paquetes adicionales:  
 libblas3 liblinear3 liblua5.3-0  
Paquetes sugeridos:  
 liblinear-tools liblinear-dev ndiff  
Se instalarán los siguientes paquetes NUEVOS:  
 libblas3 liblinear3 liblua5.3-0 nmap  
0 actualizados, 4 nuevos se instalarán, 0 para eliminar y 14 no actualizados.  
Se necesita descargar 5.467 kB de archivos.  
Se utilizarán 25,0 MB de espacio de disco adicional después de esta operación.  
¿Desea continuar? [S/n]
```

Ilustración 45: Nmap - Instalamos nmap

Tras la correcta instalación de la herramienta: **nmap**, podremos pasar al siguiente apartado.

```
¿Desea continuar? [S/n] s
Des:1 http://es.archive.ubuntu.com/ubuntu bionic/main amd64 libblas3 amd64 3.7.1-4ubuntu1 [140 kB]
Des:2 http://es.archive.ubuntu.com/ubuntu bionic/main amd64 liblinear3 amd64 2.1.0+dfsg-2 [89,3 kB]
Des:3 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 liblua5.3-0 amd64 5.3.3-1ubuntu0.18.04.1 [115 kB]
Des:4 http://es.archive.ubuntu.com/ubuntu bionic/main amd64 nmap amd64 7.60-1ubuntu5 [5.174 kB]
Descargados 5.467 kB en 3s (1.920 kB/s)
Seleccionando el paquete libblas3:amd64 previamente no seleccionado.
(Leyendo la base de datos ... 66977 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../libblas3_3.7.1-4ubuntu1_amd64.deb ...
Desempaquetando libblas3:amd64 (3.7.1-4ubuntu1) ...
Seleccionando el paquete liblinear3:amd64 previamente no seleccionado.
Preparando para desempaquetar .../liblinear3_2.1.0+dfsg-2_amd64.deb ...
Desempaquetando liblinear3:amd64 (2.1.0+dfsg-2) ...
Seleccionando el paquete liblua5.3-0:amd64 previamente no seleccionado.
Preparando para desempaquetar .../liblua5.3-0_5.3.3-1ubuntu0.18.04.1_amd64.deb ...
Desempaquetando liblua5.3-0:amd64 (5.3.3-1ubuntu0.18.04.1) ...
Seleccionando el paquete nmap previamente no seleccionado.
Preparando para desempaquetar .../nmap_7.60-1ubuntu5_amd64.deb ...
Desempaquetando nmap (7.60-1ubuntu5) ...
Configurando libblas3:amd64 (3.7.1-4ubuntu1) ...
update-alternatives: utilizando /usr/lib/x86_64-linux-gnu/blas/libblas.so.3 para proveer /usr/lib/x86_64-linux-gnu/libblas.so.3 (libblas.so.3-x86_64-linux-gnu) en modo automático
Configurando liblinear3:amd64 (2.1.0+dfsg-2) ...
Configurando liblua5.3-0:amd64 (5.3.3-1ubuntu0.18.04.1) ...
Configurando nmap (7.60-1ubuntu5) ...
Procesando disparadores para man-db (2.8.3-2ubuntu0.1) ...
Procesando disparadores para libc-bin (2.27-3ubuntu1) ...
root@mishodan-sonda:/home/soc#
```

Ilustración 46: Nmap - Instalación completada

NOTA: Se quiere mostrar el contenido del archivo por el que la herramienta: **nmap**, categoriza los puertos encontrados en el proceso de enumeración.

Dicho archivo se encuentra ubicado en: **/usr/share/nmap/nmap-services**.

```
soc@mishodan-sonda:/usr/share/nmap$ head -n 30 nmap-services
# THIS FILE IS GENERATED AUTOMATICALLY FROM A MASTER - DO NOT EDIT.
# EDIT /nmap-private-dev/nmap-services-all IN SVN INSTEAD.
# Well Known service port numbers -- mode: fundamental; -*-*
# From the Nmap Security Scanner ( https://nmap.org/ )
#
# $Id: nmap-services 36906 2017-07-31 22:29:24Z dmiller $
#
# Derived from IANA data and our own research
#
# This collection of service data is (C) 1996-2011 by Insecure.Com
# LLC. It is distributed under the Nmap Open Source license as
# provided in the COPYING file of the source distribution or at
# https://svn.nmap.org/nmap/COPYING . Note that this license
# requires you to license your own work under a compatible open source
# license. If you wish to embed Nmap technology into proprietary
# software, we sell alternative licenses (contact sales@insecure.com).
# Dozens of software vendors already license Nmap technology such as
# host discovery, port scanning, OS detection, and version detection.
# For more details, see https://nmap.org/book/man-legal.html
#
# Fields in this file are: Service name, portnum/protocol, open-frequency, optional comments
#
tcpmux 1/tcp    0.001995      # TCP Port Service Multiplexer [rfc-1078] | TCP Port Service Multiplexer
tcpmux 1/udp    0.001236      # TCP Port Service Multiplexer
compressnet 2/tcp   0.000013      # Management Utility
compressnet 2/udp   0.001845      # Management Utility
compressnet 3/tcp   0.001242      # Compression Process
compressnet 3/udp   0.001532      # Compression Process
unknown 4/tcp   0.000477
rje    5/tcp   0.000000      # Remote Job Entry
soc@mishodan-sonda:/usr/share/nmap$ _
```

Ilustración 47: Nmap - Archivo de catalogación de servicios de nmap



El contenido del archivo, tal y como los autores de la herramienta comentan, es una mezcla entre la experiencia de los mismos y la lista declarada por el organismo internacional: **IANA**. Este organismo internacional es el más importante a nivel mundial ya que es tal y como describe en su web: *iana.org*.

"The global coordination of the DNS Root, IP addressing, and other Internet protocol resources is performed as the Internet Assigned Numbers Authority (IANA) functions."

Para obtener el listado de la información compartida por dicho organismo, visitar la siguiente URL: <https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>



AI.4. Instalación de la herramienta: Logstash

Tal y como comentamos en el apartado: A.3.A.4, es muy importante para que funcione **Logstash** haber instalado previamente “java”. Por lo tanto, los pasos a seguir para la instalación de **Logstash**, son:

- 1.- Ejecutamos el comando: **apt-get install openjdk-8-jre openjdk-8-jdk**, que permitirá la instalación del aplicativo **JAVA**.

```
root@mishodan-sonda:/home/soc# apt-get install openjdk-8-jdk openjdk-8-jre
```

Ilustración 48: Logstash - Instalación JAVA

```
Configurando openjdk-8-jdk:amd64 (8u242-b08-0ubuntu3~18.04) ...
update-alternatives: utilizando /usr/lib/jvm/java-8-openjdk-amd64/bin/appletviewer para proveer /usr/bin/appletviewer (appletviewer) en modo automático
update-alternatives: utilizando /usr/lib/jvm/java-8-openjdk-amd64/bin/jconsole para proveer /usr/bin/jconsole (jconsole) en modo automático
Procesando disparadores para libgdk-pixbuf2.0-0:amd64 (2.36.11-2) ...
Procesando disparadores para libc-bin (2.27-9ubuntu1) ...
root@mishodan-sonda:/home/soc# _
```

Ilustración 49: Logstash - Instalación JAVA completada

- 2.- A partir de aquí, empezaremos los pasos para instalar la herramienta que realmente nos ocupa. Para ello, comenzamos ejecutando el comando: **wget https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt key add**, que descargará la clave pública que cargará dentro de la aplicación apt. Con ello, podremos conectarnos al repositorio habilitado por el creador de “elasticsearch” y descargar lo necesario para su instalación.

```
root@mishodan-sonda:/home/soc# rm GPG-KEY-elasticsearch
root@mishodan-sonda:/home/soc# wget https://artifacts.elastic.co/GPG-KEY-elasticsearch
--2020-03-06 13:04:40-- https://artifacts.elastic.co/GPG-KEY-elasticsearch
Resolving artifacts.elastic.co (artifacts.elastic.co)... 151.101.134.222, 2a04:4e42:1f::734
Connecting to artifacts.elastic.co (artifacts.elastic.co)|151.101.134.222|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1768 (1,7K) [application/pgp-keys]
Saving to: ‘GPG-KEY-elasticsearch’

GPG-KEY-elasticsearch    100%[=====]   1,73K  --.-KB/s   in 0s

2020-03-06 13:04:40 (145 MB/s) - ‘GPG-KEY-elasticsearch’ saved [1768/1768]
root@mishodan-sonda:/home/soc# _
```

Ilustración 50: Logstash - Descarga clave pública acceso a repositorio Logstash

```
root@mishodan-sonda:/home/soc# apt-key add GPG-KEY-elasticsearch
OK
```

Ilustración 51: Logstash - Inclusión clave pública en nuestro sistema

- 3.- Ejecutamos el comando: **apt-get install apt-transport-https**, que permitirá el uso por parte del comando **apt** de conexiones **https**.



```

root@mishodan-sonda:/home/soc# apt-get install apt-transport-https
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  apt-transport-https
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 14 no actualizados.
Se necesita descargar 1.692 B de archivos.
Se utilizarán 153 KB de espacio de disco adicional después de esta operación.
Des:1 http://es.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 apt-transport-https all 1.6.
12 [1.692 B]
Descargados 1.692 B en 0s (4.021 B/s)
Seleccionando el paquete apt-transport-https previamente no seleccionado.
(Leyendo la base de datos ... 83774 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../apt-transport-https_1.6.12_all.deb ...
Desempaquetando apt-transport-https (1.6.12) ...
Configurando apt-transport-https (1.6.12) ...
root@mishodan-sonda:/home/soc#

```

Ilustración 52: Logstash - Instalación apt-transport-https

4.- Ejecutamos el comando:

echo "deb <https://artifacts.elastic.co/packages/7.x/apt> stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list, que permitirá incluir dentro de nuestro fichero de fuentes el repositorio de ElasticSearch.

```

root@mishodan-sonda:/home/soc# echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main"
| tee -a /etc/apt/sources.list.d/elastic-7.x.list
deb https://artifacts.elastic.co/packages/7.x/apt stable main

```

Ilustración 53: Logstash - Incluimos los repositorios de logstash en nuestro fichero de fuentes

5.- Ejecutamos el comando: **apt-get install logstash**, con el que instalaremos definitivamente la herramienta.

```

root@mishodan-sonda:/home/soc# apt-get update && apt-get install logstash
Obj:1 http://es.archive.ubuntu.com/ubuntu bionic InRelease
Obj:2 http://es.archive.ubuntu.com/ubuntu bionic-updates InRelease
Des:3 https://artifacts.elastic.co/packages/7.x/apt stable InRelease [7.123 B]
Obj:4 http://es.archive.ubuntu.com/ubuntu bionic-backports InRelease
Obj:5 http://es.archive.ubuntu.com/ubuntu bionic-security InRelease
Des:6 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 Packages [30,7 KB]
Descargados 30,7 KB en 1s (29,8 KB/s)
Leyendo lista de paquetes... Hecho
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  logstash
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 14 no actualizados.
Se necesita descargar 174 MB de archivos.
Se utilizarán 305 MB de espacio de disco adicional después de esta operación.
Des:1 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 logstash all 1:7.6.1-1 [174 MB]
Descargados 174 MB en 1min 55s (1.513 KB/s)
Seleccionando el paquete logstash previamente no seleccionado.
(Leyendo la base de datos ... 83778 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../logstash_1%3a7.6.1-1_all.deb ...
Desempaquetando logstash (1:7.6.1-1) ...
Configurando logstash (1:7.6.1-1) ...
Using provided startup.options file: /etc/logstash/startup.options
/usr/share/logstash/vendor/bundle/jruby/2.5.0/gems/pleaserun-0.0.30/lib/pleaserun/platform/base.rb:1
12: warning: constant ::Fixnum is deprecated
Successfully created system startup script for Logstash

```

Ilustración 54: Logstash - Instalación Logstash



6.- Tras terminar la instalación y desarrollar el archivo de configuración visto [aquí](#). Se procede a crear el siguiente archivo que nos permitirá mantener el servicio activo siempre que se reinicie la máquina.

```
GNU nano 2.9.3                                         Lanzador-LogStash.sh
#!/bin/bash
#Lanzamos Logstash
nohup /usr/share/logstash/bin/logstash -f /home/Sonda/csv.conf
```

Ilustración 55: LogStash - Lanzador-LogsStash.sh

Por lo tanto los pasos a seguir son:

6.1.- Crear el archivo: **Lanzador-LogSatash.sh** a la ubicación: **/etc/init.d/**, y establecer la configuración anterior.

6.2.- Cambiar el propietario del archivo, mediante el comando: **chown root:root Lanzador-LogStash.sh** :

6.3.- Cambiar los permisos, para permitir la ejecución del archivo: **Lanzador-LogStash.sh**, mediante el comando: **chmod 755 Lanzador-LogStash.sh**

6.4.- Introducir el archivo: **rc.local**, dentro de la ubicación: **/etc**

```
GNU nano 2.9.3                                         rc.local
#!/bin/bash
sh /etc/init.d/Lanzador-LogStash.sh
exit 0
```

Ilustración 56: LogStash - rc.local

IMPORTANTE: Dicho archivo tiene que tener como propietario al usuario: **root**, y, además debe de tener dicho usuario **permiso de ejecución**

A1.5. Bastionado del servicio: SSH

Para bastionar el servicio: SSH, de nuestra sonda de enumeración, utilizaremos como referencia la guía del CCN-CERT número 665, publicada en Octubre de 2014.

Para bastionar nuestro servicio SSH, tenemos que acudir a la configuración del mismo. El archivo de configuración reside en la siguiente ubicación: **/etc/ssh/ssdh_config**

```
soc@mishodan-sonda:/etc/ssh$ ls -la
total 596
drwxr-xr-x  2 root root  4096 mar  6 11:19 .
drwxr-xr-x  97 root root  4096 mar 25 14:22 ..
-rw-r--r--  1 root root 553122 mar  4  2019 moduli
-rw-r--r--  1 root root  1580 mar  4  2019 ssh_config
-rw-r--r--  1 root root  3291 mar  6 11:19 sshd_config
```

Ilustración 57: SSH - Ubicación de los archivos de configuración

Aunque lo mejor sería configurar nuestro servicio SSH para que trabajase con un sistema RADIUS o trabajará con certificados asimétricos (dos certificados, uno privado que residirá en el ordenador desde donde nos queramos conectar, y otro público, que residiría en nuestra sonda de enumeración).

Nosotros vamos a permitir el acceso al servicio mediante el uso de usuario/contraseña.

Las medidas a utilizar para el bastionado de nuestro servicio serán:

1. Estableceremos un banner SSH, que muestre un mensaje alertando de las consecuencias legales que un atacante podría tener en caso de realizar alguna acción dañina contra el sistema, tal y como se puede leer en los puntos 165, 166, 167 y 168 del manual utilizado como referencia.

Además, tal y como marca el punto 170, el banner no debe proporcionar ninguna información sobre el servicio SSH

Para ello seguiremos el proceso descrito en el punto 171, es decir:

1.1 Creamos un archivo de nombre: **banner**, que contendrá el contenido a mostrar.

1.2 Descomentamos la línea “**Banner no**” del archivo de configuración del servicio y la sustituimos por: **Banner /etc/ssh/banner**

1.3 Reiniciamos el servicio

```

root@mishodan-sonda:/etc/ssh# more banner
AVISO IMPORTANTE

El acceso no autorizado a este sistema está prohibido y será castigado por la ley
Al acceder a este sistema, el usuario acepta que sus acciones pueden ser monitorizadas y registradas

root@mishodan-sonda:/etc/ssh# cat sshd_config | grep banner
# no default banner path
Banner /etc/ssh/banner
root@mishodan-sonda:/etc/ssh# service ssh restart
root@mishodan-sonda:/etc/ssh# ssh 127.0.0.1
AVISO IMPORTANTE

El acceso no autorizado a este sistema está prohibido y será castigado por la ley
Al acceder a este sistema, el usuario acepta que sus acciones pueden ser monitorizadas y registradas
root@127.0.0.1's password: _

```

Ilustración 58: SSH - Banner

- Definimos los usuarios que pueden acceder a través de este servicio a la sonda tal y como se puede leer en el punto 176. Para ello, incluimos la línea: **AllowUsers soc**

```

root@mishodan-sonda:/etc/ssh# cat sshd_config | grep AllowUsers
AllowUsers soc
root@mishodan-sonda:/etc/ssh# ssh soc@127.0.0.1
AVISO IMPORTANTE

El acceso no autorizado a este sistema está prohibido y será castigado por la ley
Al acceder a este sistema, el usuario acepta que sus acciones pueden ser monitorizadas y registradas
soc@127.0.0.1's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-88-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

 System information as of Sun Mar 29 11:57:56 UTC 2020

 System load: 0.08           Processes:          171
 Usage of /: 27.6% of 19.56GB   Users logged in:     1
 Memory usage: 18%            IP address for ens33: 192.168.0.195
 Swap usage: 0%

Pueden actualizarse 28 paquetes.
0 actualizaciones son de seguridad.

*** Es necesario reiniciar el sistema ***
Last login: Sun Mar 29 11:58:28 2020 from 127.0.0.1
soc@mishodan-sonda:~$ 

```

Ilustración 59: SSH - Definimos los usuarios que pueden usar el servicio SSH

- Definimos el tiempo en el que la sonda esperará a que se conecte un usuario, a través del servicio SSH, tal y como se puede leer en el punto 181. Para ello, descomentamos la línea: **LoginGraceTime**, y la establecemos a: **1m**

```

root@mishodan-sonda:/etc/ssh# cat sshd_config | grep LoginGraceTime
LoginGraceTime 1m

```

Ilustración 60: SSH - Tiempo máximo para acceder

- Definimos el número máximo de intentos que podrán hacer el usuario que estén intentando acceder al servicio, tal y como se puede leer en el punto



181. Para ello, descomentamos la línea: **MaxAuthTries**, y la establecemos a: **2**

```
root@mishodan-sonda:/etc/ssh# cat sshd_config | grep MaxAuthTries  
MaxAuthTries 2
```

Ilustración 61: SSH - Máximo número de intentos de acceso

5. Definimos el número de sesiones no autenticadas (pre-login) que pueden coexistir, tal y como se puede leer en el punto 182.

NOTA: Como la guía es antigua no ha tenido en cuenta posteriores modificaciones del servicio. En posteriores versiones, se ha sustituido: **Maxstartups**, por **Maxsessions**

Por ello, descomentamos la línea: **Maxsessions**, y la establecemos a: **2**

```
root@mishodan-sonda:/etc/ssh# cat sshd_config | grep MaxSessions  
MaxSessions 2
```

Ilustración 62: SSH - Número máximo de sesiones concurrentes

6. Definimos que el usuario: root, no pueda conectarse a través de dicho servicio. Para ello, descomentamos la línea: **PermitRootLogin**, y las establecemos a: **no**

```
root@mishodan-sonda:/etc/ssh# cat sshd_config | grep PermitRootLogin  
PermitRootLogin no
```

Ilustración 63: SSH - No permitimos validación por parte de ROOT

Tras todos estos cambios, será necesario reiniciar el servicio mediante el reinicio de la sonda, o del propio servicio SSH

ANEXO II: CODIGOS COMPLETOS

AII.1. Shodan.sh

Este archivo contendrá el siguiente código:



```

1 #!/bin/bash
2
3 #Variable utilizada mientras se espera que reduzca el número de hilos lanzados de Nmap
4 durmiente=0
5 #Variable que controla el número de procesos en paralelo, esto depende de la memoria del sistema
6 hilos="15"
7
8 rangos=(10)
9
10 if [ -f /home/Sonda/ControlIP.txt ]; then
11     aa=$(cat /home/Sonda/ControlIP.txt | cut -d ":" -f 1)
12     bb=$(cat /home/Sonda/ControlIP.txt | cut -d ":" -f 2)
13     cc=$(cat /home/Sonda/ControlIP.txt | cut -d ":" -f 3)
14     dd=$(cat /home/Sonda/ControlIP.txt | cut -d ":" -f 4)
15
16     if [ $bb == "" ] || [ $cc == "" ] || [ $dd == "" ] || [ $bb -lt 0 ] || [ $bb -gt 254 ] || [ $cc -lt 0 ] || [ $cc -gt
17         254 ] || [ $dd -lt 0 ] || [ $dd -gt 254 ]; then
18         echo "Hay un problema con alguno de los valores ubicado en el fichero: ControlIP.txt" > /home/Sonda/Logs/Logs
19     else
20         while true; do
21             for b in `seq $bb 254`; do
22                 for c in `seq $cc 254`; do
23                     for d in `seq $dd 254`; do
24                         for port in `seq 1 65535`; do
25
26                             if [ $(ps -ax | grep "nmap" | wc -l) -lt $hilos ]; then
27                                 echo ${rangos[0]}.$b.$c.$d":"$port > /home/Sonda/Logs/LogsEscaneos.logs
28                                 $(nmap -Pn -sV -open -p $port ${rangos[0]}.$b.$c.$d -oX
29                                     /home/Sonda/Resultados/${rangos[0]}.$b.$c.$d-$port.nmap 1> /dev/null 2> /dev/null &
30                             else
31                                 while [ $(ps -ax | grep "nmap" | wc -l) -ge $hilos ]; do
32                                     durmiente=$((durmiente + 1))
33                                     done
34                                 fi
35
36
37
38

```

Ilustración 64: Códigos - Shodan.sh - 1

```
39          done
40      echo ${rangos[0]}."$b"."$c"."$d > /home/Sonda/ControlIP.txt
41      done
42      dd=0
43      done
44      cc=0
45      done
46      bb=0
47      done
48  fi
49 else
50     echo "No existe el fichero que permite el comienzo del script" > /home/Sonda/Logs/Logs-AccesosFicheros
51 fi
52
```

Ilustración 65: Códigos - Shodan.sh -2

All.2. ParseoShodan.sh

Este archivo contendrá el siguiente código:

```

1  #!/bin/bash
2
3  while true; do
4
5      #Variables que almacenaran la fecha y hora del sistema en la que se parsea la informacion
6      #y que sera añadida a la base de datos
7      fecha=$(date +%Y-%m-%d)
8      hora=$(date +%H:%M:%S)
9
10     for archivo in $(ls /home/Sonda/Resultados/*.nmap)
11     do
12         ip=$(echo $archivo | cut -d "-" -f 1 | cut -d "/" -f 5)
13
14         #Información almacenada a modo de Log
15         echo $fecha";"$archivo";"$ip > /home/Sonda/Logs/LogsParseo.logs
16
17         #Se extrae la información que se almacenará en la base de datos.
18         #Dicha información se almacenara temporalmente en un archivo que posteriormente se importara
19         #a la base de datos.
20         grep "state=open\"" $archivo | awk -F "protocol=" '{print $2}' | awk -F "portid=" '{print $1" "$2}'
21         # | cut -d ">" -f 1,3 | awk -F "><service name=" '{print $1" "$2}' | awk -F "product=" '{print $1"
22         "$2}' | awk -F "version=" '{print $1" "$2}' | cut -d "=" -f 1 | sed 's//,/g' | cut -d "," -f 2,4,6,8
23         ,10 | sed "s/^/NULL,$fecha $hora,$ip,/" >> /home/Sonda/cit_resources.csv
24
25         rm $archivo
26     done
27
28     sleep 60
29
30 done

```

Ilustración 66: Códigos - ParseoShodan.sh

All.3. ControlEstado.sh

Este archivo contendrá el siguiente código:

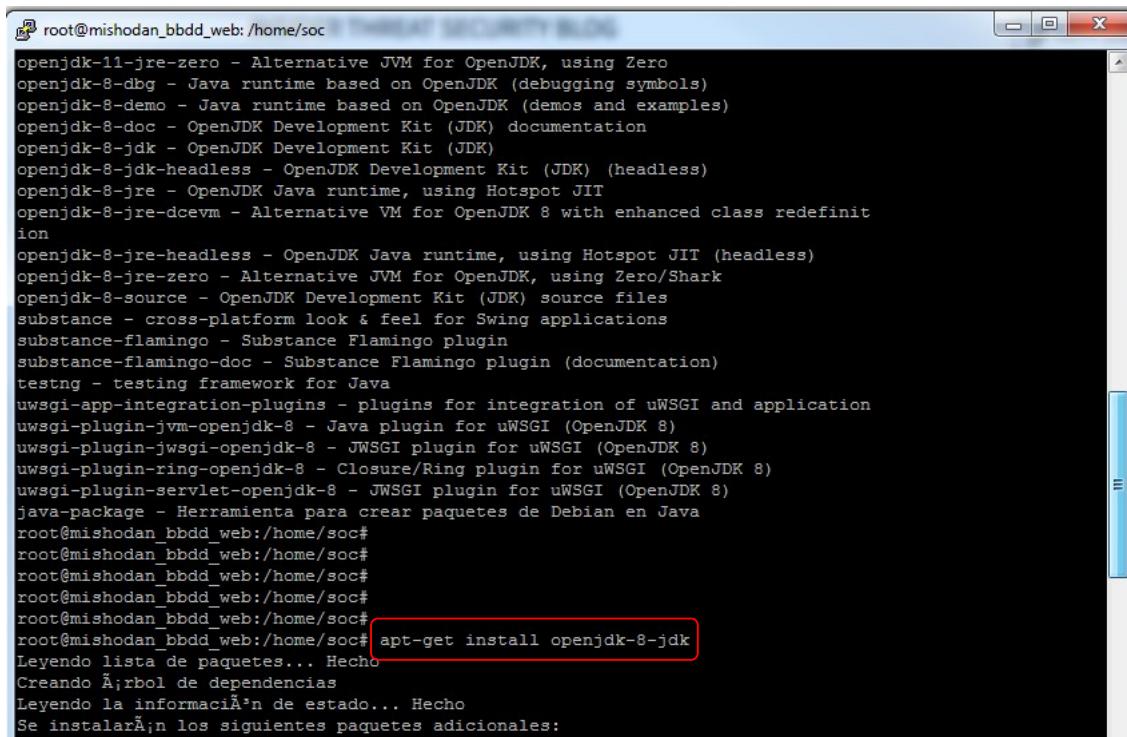
```
1  #!/bin/bash
2
3  while true; do
4
5    fecha=$(date +%Y-%m-%d)
6    hora=$(date +%H:%M:%S)
7
8    cnumerico=$(ps -aux | grep "nmap" | wc -l)
9
10   #echo "Control numerico = "$cnumerico
11
12   if [[ $cnumerico -eq 1 ]]; then
13     #Información almacenada a modo de Log
14     echo $fecha";"$hora"--No se encontro ningun proceso 'nmap' activo, se procede al reinicio del
15     servidor" > /home/Sonda/Logs/LogsControlEstado.logs
16
17     estado=$(shutdown -r now &)
18     #echo "Variable estado="$estado
19   else
20     sleep 60
21   fi
22 done
```

Ilustración 67: Códigos - ControlEstado.sh

ANEXO III: PROCESO DE INSTALACIÓN DE LOS SERVIDORES WEB, BBDD Y PROXY-WEB

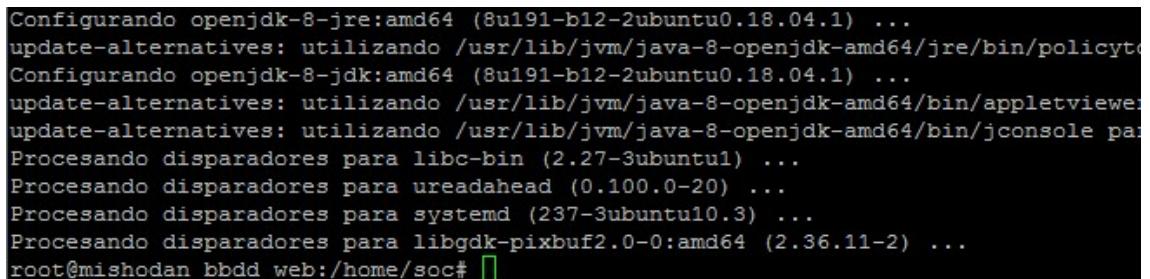
AIII.1. Instalación de ElasticSearch

Antes de nada, se procede a instalar Java en su versión *OpenJdk*



```
root@mishodan_bbdd_web: /home/soc
openjdk-11-jre-zero - Alternative JVM for OpenJDK, using Zero
openjdk-8-dbg - Java runtime based on OpenJDK (debugging symbols)
openjdk-8-demo - Java runtime based on OpenJDK (demos and examples)
openjdk-8-doc - OpenJDK Development Kit (JDK) documentation
openjdk-8-jdk - OpenJDK Development Kit (JDK)
openjdk-8-jdk-headless - OpenJDK Development Kit (JDK) (headless)
openjdk-8-jre - OpenJDK Java runtime, using Hotspot JIT
openjdk-8-jre-dcevm - Alternative VM for OpenJDK 8 with enhanced class redefinition
openjdk-8-jre-headless - OpenJDK Java runtime, using Hotspot JIT (headless)
openjdk-8-jre-zero - Alternative JVM for OpenJDK, using Zero/Shark
openjdk-8-source - OpenJDK Development Kit (JDK) source files
substance - cross-platform look & feel for Swing applications
substance-flamingo - Substance Flamingo plugin
substance-flamingo-doc - Substance Flamingo plugin (documentation)
testng - testing framework for Java
uwsgi-app-integration-plugins - plugins for integration of uWSGI and application
uwsgi-plugin-jvm-openjdk-8 - Java plugin for uWSGI (OpenJDK 8)
uwsgi-plugin-jwsgi-openjdk-8 - JWSGI plugin for uWSGI (OpenJDK 8)
uwsgi-plugin-ring-openjdk-8 - Closure/Ring plugin for uWSGI (OpenJDK 8)
uwsgi-plugin-servlet-openjdk-8 - JWSGI plugin for uWSGI (OpenJDK 8)
java-package - Herramienta para crear paquetes de Debian en Java
root@mishodan_bbdd_web:/home/soc#
root@mishodan_bbdd_web:/home/soc#
root@mishodan_bbdd_web:/home/soc#
root@mishodan_bbdd_web:/home/soc#
root@mishodan_bbdd_web:/home/soc#
root@mishodan_bbdd_web:/home/soc# apt-get install openjdk-8-jdk
Leyendo lista de paquetes... Hecho
Creando Árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
```

Ilustración 68: ElasticSearch - Instalación de Java



```
Configurando openjdk-8-jre:amd64 (8u191-b12-2ubuntu0.18.04.1) ...
update-alternatives: utilizando /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/policycy...
Configurando openjdk-8-jdk:amd64 (8u191-b12-2ubuntu0.18.04.1) ...
update-alternatives: utilizando /usr/lib/jvm/java-8-openjdk-amd64/bin/appletviewer...
update-alternatives: utilizando /usr/lib/jvm/java-8-openjdk-amd64/bin/jconsole pa...
Procesando disparadores para libc-bin (2.27-3ubuntu1) ...
Procesando disparadores para ureadahead (0.100.0-20) ...
Procesando disparadores para systemd (237-3ubuntu10.3) ...
Procesando disparadores para libgdk-pixbuf2.0-0:amd64 (2.36.11-2) ...
root@mishodan_bbdd_web:/home/soc#
```

Ilustración 69: ElasticSearch - Tras la llamada de instalación

Tras lo cual, ya se puede proceder a la instalación del Elasticsearch, para lo cual se debe:

- 1.- Descargar desde: <https://www.elastic.co/downloads/elasticsearch/> el archivo con extensión: .deb
- 2.- Tras la descarga se procede a lanzar el comando: **dpkg -i elasticsearch**

```

root@mishodan_bbdd_web:/home/soc# dpkg -i elasticsearch-6.6.1.deb
Seleccionando el paquete elasticsearch previamente no seleccionado.
(Leyendo la base de datos ... 118371 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar elasticsearch-6.6.1.deb ...
/usr/bin/java
Creating elasticsearch group... OK
Creating elasticsearch user... OK
Desempaquetando elasticsearch (6.6.1) ...
Configurando elasticsearch (6.6.1) ...
Created elasticsearch keystore in /etc/elasticsearch
Procesando disparadores para systemd (237-3ubuntu10.13) ...
Procesando disparadores para ureadahead (0.100.0-20) ...
root@mishodan_bbdd_web:/home/soc#

```

Ilustración 70: ElasticSearch - Instalación de ElasticSearch

3.- Tras la instalación se procede a configurar el servicio, para lo cual se edita el archivo: **/etc/elasticsearch/elasticsearch.yml**. Dentro del archivo se crean las siguientes líneas:

```

Network.host: 0.0.0.0
Http.port: 9200
Discovery.type: single-node

```

```

GNU nano 2.9.3                               /etc/elasticsearch/elasticsearch.yml

path.data: /var/lib/elasticsearch
#
# Path to log files:
#
path.logs: /var/log/elasticsearch
#
# ----- Memory -----
#
# Lock the memory on startup:
#
#bootstrap.memory_lock: true
#
# Make sure that the heap size is set to about half the memory available
# on the system and that the owner of the process is allowed to use this
# limit.
#
# Elasticsearch performs poorly when the system is swapping the memory.
#
# ----- Network -----
#
# Set the bind address to a specific IP (IPv4 or IPv6):
#
network.host: 0.0.0.0
#
# Set a custom port for HTTP:
#
http.port: 9200
#
# For more information, consult the network module documentation.
#
# ----- Discovery -----

```

Ilustración 71: ElasticSearch - Configuración de ElasticSearch - Parte 1

```

GNU nano 2.9.3                               /etc/elasticsearch/elasticsearch.yml

#
# Pass an initial list of hosts to perform discovery when this node is started:
# The default list of hosts is ["127.0.0.1", "[::1]"]
#
#discovery.seed_hosts: ["host1", "host2"]
#
discovery.type: single-node
#
# Bootstrap the cluster using an initial set of master-eligible nodes:
#
#cluster.initial_master_nodes: ["node-1", "node-2"]
#
# For more information, consult the discovery and cluster formation module documentation.
#
# ----- Gateway -----
#
# Block initial recovery after a full cluster restart until N nodes are started:
#
#gateway.recover_after_nodes: 3
#
# For more information, consult the gateway module documentation.
#
# ----- Various -----
#
# Require explicit names when deleting indices:
#
#action.destructive_requires_name: true

```

Ilustración 72: ElasticSearch - Configuración de ElasticSearch - Parte 2

4.- Tras salvar los anteriores datos, se procede a iniciar el servicio.

```

root@mishodan_bbdd_web:/etc/elasticsearch# service elasticsearch start
root@mishodan_bbdd_web:/etc/elasticsearch# service elasticsearch status
  ● elasticsearch.service - Elasticsearch
      Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; disabled; vendor preset: enabled)
      Active: active (running) since Thu 2019-02-21 08:04:28 UTC; 3s ago
        Docs: http://www.elastic.co
     Main PID: 3654 (java)
        Tasks: 21 (limit: 4664)
       CGroup: /system.slice/elasticsearch.service
           └─3654 /usr/bin/java -Xms1g -Xmx1g -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFra
               ├─3756 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/bin/controller
               └─3757 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/bin/controller

feb 21 08:04:28 mishodan_bbdd_web systemd[1]: Started Elasticsearch.

```

Ilustración 73: ElasticSearch - Iniciando el servicio.

5.- Configurar el sistema para levantar el servicio siempre que se reinicie la máquina.

```

GNU nano 2.9.3

#!/bin/bash

#Lanzamos Elasticsearch
nohup service elasticsearch start &

```

Ilustración 74: ElasticSearch - Lanzador-EntornoELK.sh

Por lo tanto los pasos a seguir son:

5.1.- Crear el archivo: **Lanzador-EntornoELK.sh** a la ubicación: **/etc/init.d/**, y establecer la configuración anterior.

5.2.- Cambiar el propietario del archivo, mediante el comando: **chown root:root Lanzador-EntornoELK.sh**

5.3.- Cambiar los permisos, para permitir la ejecución del archivo: **Lanzador-EntornoELK.sh**, mediante el comando: **chmod 755 Lanzador-EntornoELK.sh**

5.4.- Introducir el archivo: **rc.local**, dentro de la ubicación: **/etc**



```
GNU nano 2.9.3                               /etc/rc.local
#!/bin/bash
sh /etc/init.d/Lanzador-EntornoELK.sh
exit 0
```

Ilustración 75: ElasticSearch - rc.local

IMPORTANTE: Dicho archivo tiene que tener como propietario al usuario: **root**, y, además debe de tener dicho usuario **permiso de ejecución**

AIII.2. Instalación de Kibana

Tras lo cual, ya se puede proceder a la instalación del Elasticsearch, para lo cual se debe:

1.- Descargar desde: <https://www.elastic.co/downloads/kibana>, el archivo con extensión: **.deb**

2.- Tras la descarga se procede a lanzar el comando: **dpkg -i kibana-6.6.1-amd64.deb**

```
root@mishodan_bbdd_web:/home/soc# dpkg -i kibana-6.6.1-amd64.deb
Seleccionando el paquete kibana previamente no seleccionado.
(Leyendo la base de datos ... 118781 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar kibana-6.6.1-amd64.deb ...
Desempaquetando kibana (6.6.1) ...
Configurando kibana (6.6.1) ...
Procesando disparadores para systemd (237-3ubuntu10.13) ...
Procesando disparadores para ureadahead (0.100.0-20) ...
root@mishodan_bbdd_web:/home/soc# ls
```

Ilustración 76: Kibana - Instalación de Kibana

3.- Tras la instalación se procede a configurar el servicio, para lo cual se edita el archivo: **/etc/kibana/kibana.yml**. Dentro del archivo se crean las siguientes líneas:

```
server.port: 5601
server.host: localhost
server.name: "Kibana-MiShodan"
elasticsearch.hosts: ["http://localhost:9200"]
```

```

GNU nano 2.9.3                               /etc/kibana/kibana.yml

# Kibana is served by a back end server. This setting specifies the port to use.
server.port: 5601

# Specifies the address to which the Kibana server will bind. IP addresses and host names are both valid values.
# 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: "localhost"

# 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

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

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

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

# When this setting's value is true Kibana uses the hostname specified in the server.host
# setting. When the value of this setting is false, Kibana uses the hostname of the host
# that connects to this Kibana instance.
#elasticsearch.preserveHost: true

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

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

```

Ilustración 77: Kibana - Configuración del Kibana

4.- Tras salvar los anteriores datos, se procede a iniciar el servicio

```

root@mishodan_bbdd_web:/etc/kibana# service kibana start
root@mishodan_bbdd_web:/etc/kibana# service kibana status
● Kibana.service - Kibana
    Loaded: loaded (/etc/systemd/system/kibana.service; disabled; vendor preset: enabled)
      Active: active (running) since Thu 2019-02-21 10:37:48 UTC; 3s ago
        Main PID: 1799 (node)
          Tasks: 11 (limit: 4662)
         CGroup: /system.slice/kibana.service
                   └─ââ1799 /usr/share/kibana/bin/.../node/bin/node --no-warnings --max-http-header-  

feb 21 10:37:48 mishodan_bbdd_web systemd[1]: Started Kibana.

```

Ilustración 78: Kibana - Iniciando el servicio.

5.- Configurar el sistema para levantar el servicio siempre que se reinicie la máquina.

```

GNU nano 2.9.3

#!/bin/bash

#Lanzamos ElasticSearch
nohup service elasticsearch start &

#Lanzamos Kibana
nohup service kibana start &

```

Ilustración 79: Kibana - Lanzador-EntornoELK.sh

Por lo tanto los pasos a seguir son:

5.1.- Crear el archivo: **Lanzador-EntornoELK.sh** a la ubicación: **/etc/init.d/**, y establecer la configuración anterior.

5.2.- Cambiar el propietario del archivo, mediante el comando: **chown root:root Lanzador-EntornoELK.sh** :

5.3.- Cambiar los permisos, para permitir la ejecución del archivo: **Lanzador-EntornoELK.sh**, mediante el comando: **chmod 755 Lanzador-EntornoELK.sh**

5.4.- Introducir el archivo: **rc.local**, dentro de la ubicación: **/etc**

```
GNU nano 2.9.3                                     /etc/rc.local

#!/bin/bash
sh /etc/init.d/Lanzador-EntornoELK.sh
exit 0
```

Ilustración 80: Kibana - rc.local

IMPORTANTE: Dicho archivo tiene que tener como propietario al usuario: **root**, y, además debe de tener dicho usuario **permiso de ejecución**

AIII.3. Instalación de Nginx

Para instalar Nginx, se necesita hacer [\[13\]](#):

- 1.- Se instala Nginx y las herramientas de apache, mediante: apt-get install nginx apache2-utils

```
root@mishodan_bbdd_web:/home/soc# apt-get install nginx apache2-utils
Leyendo lista de paquetes... Hecho
Creando Árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libapr1 libaprutil1 libgd3 libnginx-mod-http-geoip libnginx-mod-http-image-filter
  libnginx-mod-stream libwebp6 nginx-common nginx-core
Paquetes sugeridos:
  libgd-tools fcgiwrap nginx-doc ssl-cert
Se instalarán los siguientes paquetes NUEVOS:
  apache2-utils libapr1 libaprutil1 libgd3 libnginx-mod-http-geoip libnginx-mod-http-
  libnginx-mod-mail libnginx-mod-stream libwebp6 nginx nginx-common nginx-core
0 actualizados, 13 nuevos se instalarán, 0 para eliminar y 107 no actualizados.
Se necesita descargar 1.161 kB de archivos.
Se utilizarán 3.866 kB de espacio de disco adicional despues de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://archive.ubuntu.com/ubuntu bionic/main amd64 libapr1 amd64 1.6.3-2 [90,
```

Ilustración 81: Nginx - Instalación de Nginx y apache2-utils

- 2.- Tras la instalación anterior, se pasa a configurar el usuario y la contraseña con la que la solicitud del Nginx se transmitirá al Kibana. Para ello se ejecutara el comando: **htpasswd -c /etc/nginx/htpasswd.kibana <usuario>**

En nuestro caso utilizaremos los siguientes datos:

Usuario: **Adminkibana**
Contraseña: **k1b4n1t4.-**

- 3.- Se configura el servicio para tratará la redirección, para lo cual:

3.1.- Se ejecuta el siguiente comando que permite crear el archivo donde irá la configuración: **echo > /etc/nginx/sites-available/default**

3.2.- Se introduce la siguiente configuración

```
GNU nano 2.9.3                               sites-available/default

#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration
#
server {
    listen 80 default_server;
    server_name 192.168.88.129;

    auth_basic "Acesso a Kibana";
    auth_basic_user_file /etc/nginx/htpasswd.kibana;

    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        #try_files $uri $uri/ =404;
        proxy_pass http://localhost:5601;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }

    # pass PHP scripts to FastCGI server
    #
}
```

Ilustración 82: Nginx - Configuración utilizada en Nginx

- 4.- Ahora arrancaremos Nginx y configuraremos que arranque automáticamente al reiniciar el servidor, mediante los comandos:

Systemctl start nginx
systemctl enable nginx

BIBLIOGRAFÍA

- [0] https://store.vmware.com/store/vmwde/es_ES/DisplayProductDetailsPage/
- [1] <https://ubuntu.com/download/server>
- [2] <https://nmap.org>
- [3] <https://www.elastic.co/es/what-is/elk-stack>
- [4] <https://www.nginx.com/>
- [5] <https://obralibre.wordpress.com/2016/02/05/projectlibre-ingreso-de-datos-tareas/>
- [6] https://es.wikipedia.org/wiki/Classless_Inter-Domain_Routing
- [7] <http://jvallejo.epv.uniovi.es/wordpress/2015/04/08/arranque-automatico-de-xampp-en-linux/>
- [8] <https://arenlasysadmin.wordpress.com/2013/05/05/ejecutar-script-arranque-linux/>
- [9] <https://stackoverflow.com/questions/58022783/elasticsearch-error-bootstrap-checks-failed-binding-non-loopback-address>
- [10] <https://www.elastic.co/guide/en/elasticsearch/reference/current/network.host.html>
- [11] <https://www.elastic.co/guide/en/elasticsearch/reference/current/modules-network.html#network-interface-values>
- [12] <https://www.sudo.ws/man/1.8.3/sudo.man.html>
- [13] <https://clouding.io/kb/como-instalar-elk-elasticsearch-logstash-y-kibana/>
- [14] <https://www.bmc.com/blogs/elasticsearch-load-csv-logstash/>