Escuela Politécnica Superior de Jaén
Máster Universitario en Seguridad Informática

UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior (Jaén)

Trabajo Fin de Máster

AUDITORÍA DE HACKING ÉTICO SOBRE UNA PLATAFORMA ELASTIC AISLADA

Alumno/a: **Beltrán Cárdenas, Daniel**

Tutor/a: Prof. D. Miguel Ángel García Cumbreras
Dpto.: Informática

Noviembre, 2022

Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Don Miguel Ángel García Cumbreras, tutor del Proyecto Fin de Carrera titulado: Autditoría de Hacking Ético sobre una plataforma Elastic aislada, que presenta Daniel Beltrán Cárdenas, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Noviembre de 2022

El alumno:

Daniel Beltrán Cárdenas

Los tutores:

GARCIA CUMBRERAS MIGUEL ANGEL - 52871272E

Firmado digitalmente por GARCIA CUMBRERAS MIGUEL ANGEL - 52871272E
Fecha: 2022.11.21 23:30:37 +01'00'

Miguel Ángel García Cumbreras

Índice del documento

1. INTRODUCCIÓN, MOTIVACIÓN Y OBJETIVOS DEL PROYECTO. 4
1.1 Introducción y motivación. 4
1.2 Objetivo del proyecto. 4
1.3 Metodología aplicada. 5

2. ELASTICSEARCH 6
2.1 ¿ Qué es Elasticsearch ? 6
2.2 Licencias en Elasticsearch. 7

3. OPCIONES DE SEGURIDAD EN ELASTICSEARCH 11
3.1 Conceptos básicos. 11
3.2 Roles y autenticación. 11
3.3 Detecciones y alertas. 12
3.4 Opciones de seguridad en Elasticsearch. 13
3.4.1 Configuración del hashing de las contraseñas. 14
3.4.2 Acceso anónimo 14
3.4.3 Seguridad a nivel de documento y campo. 14
3.4.4 Ajustes de HTTP TLS/SSL. 14
3.4.5 Servicio de Tokens. 15
3.4.6 Ajustes de la clave de API. 15
3.4.7 Configuraciones de auditoría 15
3.4.8 Eventos de auditoría 16
3.5 Ficheros de seguridad 16

4. Configuración de TLS, roles y autenticación en elastic. 18
4.1 Roles y usuarios de elasticsearch 19
4.2- Comunicación segura entre nodos de un cluster 23
4.3- Comunicaciones seguras en el protocolo de transporte en elasticsearch 25

5.- Instalación de elasticsearch en ubuntu 27

6.- Pentesting a Elasticsearch 30
6.1 Puesta en marcha de las máquinas a utilizar 30
6.2 Hacking ético 30
6.3 Enumeración 40
6.4 Análisis de riesgos 43

7.- Vulnerabilidades y ataques a Elasticsearch 45
7.1.- Vulnerabilidades 7.x 45
7.2 Vulnerabilidades 8.x 46

8.- Mejoras en la versión 8.x 47
8.1.- Configuración inicial 47
8.2.- Comunicaciones entre clusters 47
8.3.- Resto de apartados de seguridad 48

9.- Conclusiones 49

BIBLIOGRAFÍA **50**

Índice de figuras

2.1- Figura 1: Componentes del servicio Elastic. 7
2.2- Figura 2: Licencias Elasticsearch. 8
2.2- Figura 3: Licencias Elasticsearch. 9
2.2- Figura 4: Licencias Elasticsearch. 9
4.1- Figura 5: Configuración de contraseñas. 21
4.1- Figura 6: Creación de usuarios y contraseñas. 21
4.1- Figura 7: Petición a un servicio Elastic con usuarios. 22
4.1- Figura 8: Elastic detecta que no ha habido autenticación 23
4.2- Figura 9: Generación de una CA. 24
4.2- Figura 10: Generación del certificado autofirmado 25
6.3- Figura 11: Banner Grabbing. 31
6.3- Figura 12: Comprobar si existen usuarios en el servicio. 32
6.3- Figura 13: Comprobar que paquetes tiene habilitados el servicio 32
6.3- Figura 14: Índices del servicio. 34
6.3- Figura 15: Tipo de datos de un índice consultado. 35
6.3- Figura 16: Datos dentro de un índice. 36
6.3- Figura 17: datos dentro de un índice. 36
6.3- Figura 18: Búsqueda de datos específicos mediante “search”. 38
6.3- Figura 19: Listado de índices tras la creación de uno mediante API REST. 39
6.3- Figura 20: Petición fallida debido a que es necesario autenticación. 39
6.3- Figura 21: Número de servicios escaneados con Shodan. 41
6.3- Figura 22: Banners de servicios escaneados por Shodan. 42
6.3- Figura 23: Listado de índices de un servicio expuesto a internet. 42
6.3- Figura 24: Acceso a un documento interno del servicio expuesto 44

1. INTRODUCCIÓN, MOTIVACIÓN Y OBJETIVOS DEL PROYECTO.

1.1 Introducción y motivación.

Cada vez son más las empresas y organizaciones que tienen que hacer frente a una gran cantidad de datos que reciben desde muchas partes las mismas, desde logs de servidores y análisis de rendimiento de los mismos, hasta creación documentos en la nube de la empresa así como búsquedas de los mismos documentos. Por lo que el uso de herramientas que faciliten el análisis o el simple almacenamiento de los mismos está extendiéndose.

Ahora bien, es muy importante tener herramientas poderosas para su función pero igualmente es de vital importancia que estás sean seguras, ya que estás van a manejar información sensible de la organización y una pérdida de los mismos pueda hacer caer en picado a las a organizaciones, haciéndoles perder reputación e ingresos. Es muy importante siempre que se vaya a utilizar una herramienta sea cual sea, saber que es capaz de hacer, qué opciones nos da a la hora de su uso, como mantiene la seguridad de la información, si debe estar expuesta a internet, etc. Lo más recomendable para hacer ese análisis es comprobar la propia documentación de la herramienta.

Una gran cantidad de organizaciones optan por descargar, instalar y usar, sin tener en cuenta lo que conlleva usar está herramienta, esto hace que, si esta herramienta no ha sido configurada teniendo en cuenta la seguridad de la misma, pueda ser fácilmente explotable por ciberdelincuentes.

1.2 Objetivo del proyecto.

Habiendo expuesto lo anterior, el objetivo de este proyecto es mostrar que una herramienta tan potente y extendida como puede ser Elasticsearch, si a esta no se le hace una buena configuración de la misma a la hora de su instalación, hace que pueda convertirse en un gran agujero de seguridad para la organización, ya que está probablemente tenga datos sensibles de la organización. Por tanto los objetivos serán los siguientes:

- PRIMERO. Realizar un estudio de las medidas de seguridad usadas en Elastic.
- SEGUNDO. Poner en marcha un servicio Elastic en un entorno aislado.
- TERCERO. Realizar procesos de detección y explotación sobre Elastic.

1.3 Metodología aplicada.

Para ser capaces de cumplir los objetivos propuestos, es necesario llevar a cabo una serie de acciones enfocadas a dicho propósito. A continuación en este apartado se aborda la secuencia de fases y tareas a realizar para desarrollar el trabajo:

- Estudio bibliográfico de la documentación de Elastic, para estudiar las medidas de seguridad que utiliza.
- Configuración e instalación de un servidor Elastic de en un entorno aislado con la configuración por defecto de la misma.
- Utilización técnicas de hacking como sobre la plataforma aislada para detectar y explotar distintas vías de ataque.

2. ELASTICSEARCH

En este capítulo se va a tratar de explicar lo que és elastic y su funcionamiento así como que tipo de licenciamiento permite a las organizaciones

2.1 ¿ Qué es Elasticsearch ?

Elasticsearch, a partir de ahora llamado simplemente Elastic cuando se hable del servicio en sí, es una herramienta capaz de indexar gran cantidad de información de forma distribuida, analizarla y ofrecer búsquedas complejas de la información recabada, esta herramienta es muy polivalente ya que permite: el almacenamiento de logs y análisis de los mismos, monitoreo de rendimiento de servidores o simplemente almacenamiento de documentación de una organización.

El funcionamiento de la misma hace que los datos sin procesar pasen por el servicio Elastic desde una gran cantidad de fuentes diferentes. Estos datos son parseados y normalizados antes de ser indexados. Una vez hecha la indexación los usuarios pueden hacer búsquedas complejas en el servicio sobre los datos que llegan al servicio.

Los datos obtenidos en Elastic son almacenados en un índice. Un índice es una colección de documentos relacionados entre sí, y estos son almacenados en formato JSON. Cada documento tiene una relación con un conjunto de claves, como pueden ser los nombres de los campos. Además Elastic ofrece un API REST a disposición del usuario para realizar cualquier tipo de consulta, desde ver el estado de un nodo a listar todos los documentos que existen en el almacenamiento.

Elastic tiene varios componentes que facilitan el procesamiento de los datos que entran al servicio:

- Beats: son agentes que se encargan del envío de cualquier tipo de datos a Elasticsearch o Logstash.
- Logstash: es un *pipeline* de procesamiento de datos que permite recibir como entrada datos de múltiples fuentes a la vez y pasearlos y transformarlos antes de ser indexados.
- Elasticsearch: es un servidor que provee un motor de búsqueda de texto, este componente se puede considerar un Base de datos NoSQL.

- Kibana: es una aplicación web hecha con Node.js que puede buscar, visualizar e interactuar con la información almacenada en Elasticsearch.

En la siguiente imagen se puede observar el funcionamiento del servicio Elastic, este recibe los datos de cualquier tipo, los transforma y los almacena esperando consultas.

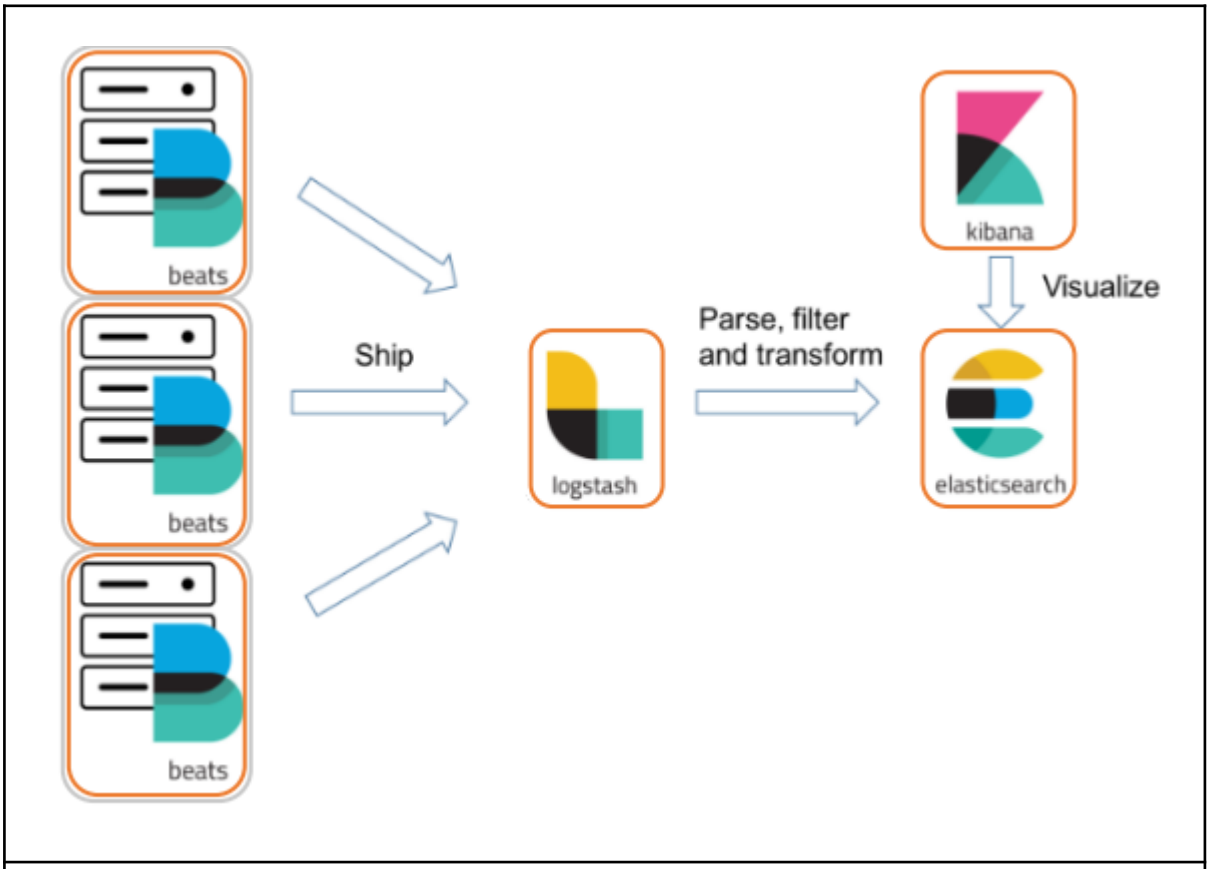


Figura 1: Componentes del servicio Elastic.

2.2 Licencias en Elasticsearch.

Elasticsearch tiene un filosofía de pago basado en los recursos, es decir, en caso de contratar una licencia de pago (existen licencias gratuitas), solo se pagará por los recursos que se están utilizando. Elasticsearch distingue entre 4 tipos de licencias: gratuitas, oro, platino y enterprise, siendo oro la más económica en precio.

Con respecto a las características entre las diferentes licencias, el aumento de las características se producen sobre todo en machine learning, seguridad y

soporte, siendo machine learning en la que más características se desbloquean. Respecto a la seguridad, la mayoría de las características básicas de seguridad vienen habilitadas desde la licencia gratuita, dejando así algunas características más avanzadas o “add-ons” a las licencias de pago.

En la siguientes imágenes se puede ver la tabla comparativa de las distintas licencias respecto a la seguridad:

	GRATUITA Y ABIERTA; BÁSICA [1,2]	ORO	PLATINO	ENTERPRISE
Seguridad del Elastic Stack				
Configuración segura	✓	✓	✓	✓
Comunicaciones encriptadas	✓	✓	✓	✓
Control de Acceso basado en roles	✓	✓	✓	✓
Control de acceso anónimo (uso compartido público)	✓	✓	✓	✓
Autenticación nativa y de archivos	✓	✓	✓	✓
Kibana Spaces	✓	✓	✓	✓
Controles de características de Kibana	✓	✓	✓	✓
Privilegios de subcaracterísticas de Kibana[8]	—	✓	✓	✓
Acuerdo de acceso previo al inicio de sesión	—	✓	✓	✓
Gestión de claves de API	✓	✓	✓	✓
Logs de auditoría de Elasticsearch	—	✓	✓	✓
Logs de auditoría de Kibana	—	✓	✓	✓
Filtrado IP	—	✓	✓	✓

Figura 2: Licencias Elasticsearch.

Autenticación de LDAP, PKI[3], Active Directory	—	✓	✓	✓
Servicio de token de Elasticsearch	—	✓	✓	✓
Inicio de sesión único (SAML, OpenID Connect, Kerberos)	—	—	✓	✓
Control de acceso basado en atributos	—	—	✓	✓
Seguridad a nivel de campo y documento	—	—	✓	✓
Realms de autenticación y autorización personalizados	—	—	✓	✓
Soporte de encryption at rest	—	—	✓	✓
Modo FIPS 140-2	—	—	✓	✓

Figura 3: Licencias Elasticsearch.

Alertas				
Alertas de Kibana	✓	✓	✓	✓
Alertas de Kibana: rastreo de contención (geofencing)	—	✓	✓	✓
Alertas de Kibana: Alerta de detección de anomalías (Machine Learning)	—	—	✓	✓
Acciones de Kibana: indexación y logging	✓	✓	✓	✓
Acciones de Kibana: correo electrónico, webhooks, Jira, Microsoft Teams, PagerDuty, Slack, Swimlane	—	✓	✓	✓
Acciones de Kibana: IBM Resilient, ServiceNow® ITSM	—	—	✓	✓
Watcher	—	✓	✓	✓

Figura 4: Licencias Elasticsearch.

Como se puede observar, la seguridad básica en elasticsearch viene preparada para su uso desde la licencia más simple, pudiendo añadir más

seguridad como por ejemplo logs de auditoría o inicio de sesión únicos adquiriendo una licencia mayor. [1]

3. OPCIONES DE SEGURIDAD EN ELASTICSEARCH

En este capítulo se va a exponer qué opciones de configuración nos ofrece Elastic en cuanto a seguridad. Elastic basa toda su configuración en un archivo YML, donde mediante palabras claves se pueden activar o desactivar unas opciones u otras.

3.1 Conceptos básicos.

Antes de entrar en la configuración de Elastic hay que explicar algunos conceptos básicos que permitirán entender mejor todas las explicaciones posteriores.

- Nodo: es un servidor que forma parte de un cluster, este almacena los datos y participa en las capacidades de indexación y búsqueda del cluster, este tiene un identificador único dentro de él
- Un cluster es una colección de nodos y contiene todos los datos sobre los que se va a buscar.
- SSL: es un tecnología estandarizada que permite cifrar los datos entre dos entidades.
- TLS: es una versión mejorada de SSL. (se explicará más adelante)
- Alerta: es un aviso que lanza Elastic cuando se cumple una serie de reglas definidas. Por ejemplo: cuando haya más de 5 intentos fallidos en la autenticación.

3.2 Roles y autenticación.

Para el acceso a los cluster, Elastic tiene un sistema interno de autenticación de usuarios y privilegios basados en roles. También soportan la autorización basada en IP. Se pueden hacer listas blancas y negras de direcciones IP o subredes para controlar el acceso al servidor. [5]

Elastic tiene definido por defecto una gran cantidad diferente de roles, que dará acceso única y exclusivamente a determinadas características a los usuarios

que tengan asignado este rol. Por ejemplo: *superuser*, este rol nos dará acceso a todas las características, este suele ser el rol más buscado a la hora de atacar un servicio.

Para la integridad de los datos lo que se realiza es la encriptación de las comunicaciones y para la auditoría de los datos se mantiene un registro activo para poder ver quien entra y quien intenta entrar.

Capas de seguridad.

1. Autenticación de usuarios.
2. Autorización de usuarios y control de acceso.
3. Autenticación de nodos/clientes y encriptación de canales.
4. Auditoría.

Por defecto las características de seguridad de Elastic están deshabilitadas cuando se tiene licencia básica. Para poder habilitar las opciones de seguridad habrá que habilitar la opción **xpack.security.enabled.**

Los ajustes xpack.security se configuran para habilitar el acceso anónimo y realizar la autenticación de mensajes, configurar la seguridad a nivel de documento y de campo, configurar los dominios, cifrar las comunicaciones con SSL y auditar los eventos de seguridad.

Todos estos ajustes pueden añadirse al archivo de configuración elasticsearch.yml, con la excepción de los ajustes de seguridad, que se añaden al almacén de claves de Elasticsearch.

3.3 Detecciones y alertas.

Elastic usa el manejo y creación de reglas para encontrar índices, eventos de origen sospechoso, y crean alertas cuando se cumplen las condiciones de una regla. Para algunas de las características expuestas, se necesita una licencia mayor a la gratuita.

Se pueden crear reglas propias o usar unas ya predefinidas, estas reglas una vez ejecutadas pueden modificarse para ajustarse mejor a lo que el usuario necesita evitando falsos positivos.

Hay dos reglas preconstruidas importantes:

- ***Endpoint security***: esta regla se encarga de activar todas las alertas que van llegando, si esta regla no está activa no se producirán alertas del tipo *Malware Prevention Alert* y *Malware Detection Alert*. [2]
- ***External alerts***: crea alertas que provienen de sistemas de terceros como puede ser suricata o snort, si no está activada, no generará alertas. [3]

Tipos de reglas:

- **Reglas personalizadas**: regla basada en una consulta, busca índices definidos y cuando coinciden varios documentos salta
- ***Machine learning***: esta regla es de una suscripción mayor a la gratuita, lo hace es basarse en el aprendizaje automático que va obteniendo.
- **Umbral**: busca índices definidos y crea una alerta cuando durante una sola ejecución el valor del campo supera el umbral.
- **Correlación de eventos**: busca índices definidos y salta cuando coinciden con una consulta de tipo *event query language* (EQL), es decir, cuando ocurren varios eventos que se han introducido como entrada.
- **Regla de coincidencia**: crea una alerta cuando un valor del campo coincide con el patrón definido. Por ejemplo, que salte una alerta cuando una ip de destino sea x.

También se pueden añadir excepciones a las reglas para así eliminar ruido

- ***Building-block rules***: permite que las reglas de alertas pequeñas no salten en la ventana principal. [4]

3.4 Opciones de seguridad en Elasticsearch.

En esta sección se va a exponer las opciones de seguridad que ofrece Elastic, estas opciones como se ha explicado anteriormente, son palabras claves que irán incluidas en el archivo de configuración elasticsearch.yml.

- ***xpack.security.enabled***: Si esta opción no está habilitada las demás opciones de seguridad tampoco podrán habilitarse.
- ***xpack.security.fips_mode.enabled***: activa el estándar de procesamiento de la información fips 140-2.

3.4.1 Configuración del hashing de las contraseñas.

- ***xpack.security.auth.password_hashing.algorithm***: algoritmo de encriptacion usado por Elastic, por defecto el algoritmo es bcrypt. (algoritmo bcrypt en https://en.wikipedia.org/wiki/Bcrypt)

3.4.2 Acceso anónimo

- ***xpack.security.authc.anonymous.authz_exception***: desecha en el proceso de autorización solicitud entrantes de forma anónima. Por defecto está en TRUE, que significa que se permiten solicitudes de forma anónima.
- ***xpack.security.auth.anonymous.roles***: se le pedirá el rol al anónimo para acceder al recurso especificado.

3.4.3 Seguridad a nivel de documento y campo.

- ***xpack.security.dls_fls.enabled***: se encarga de establecer permisos a los roles para acceder a un documento o un campo, por defecto TRUE.

3.4.4 Ajustes de HTTP TLS/SSL.

- ***xpack.security.http.ssl.enabled***: se utiliza para activar o desactivar la tecnología de comunicación de cifrado TLS/SSL. El valor por defecto es falso.
- ***xpack.security.http.ssl.supported_protocols***: protocolos soportados con versiones. Protocolos válidos: SSLv2Hello, SSLv3, TLSv1, TLSv1.1, TLSv1.2. Por defecto, TLSv1.2, TLSv1.1, TLSv1. Por defecto, el valor por defecto son TLSv1.2, TLSv1.1.
- ***xpack.security.http.ssl.client_authentication***: controla el comportamiento del servidor con respecto a la solicitud de un certificado de las conexiones del cliente. Los valores válidos son requeridos, opcional y ninguno. Requerido obliga al cliente a presentar un certificado, mientras que opcional solicita un certificado de cliente pero el cliente no está obligado a presentarlo. El valor predeterminado es ninguno.

3.4.5 Servicio de Tokens.

- ***xpack.security.auth.token.enabled***: depende de xpack.security.http.ssl.enabled, ya que si permite conexiones http, entonces esta opción no se habilitará, esta opción impide que se pueda acceder al token en una conexión.
- ***xpack.security.auth.token.passphrase***: contraseña de longitud mínima 8 para elaborar una clave criptográfica para cifrar los tokens, aunque por defecto el propio cluster genera automáticamente una clave.
- ***xpack.security.auth.token.timeout***: timeout del token, por defecto el valor es 20 minutos y máximo 1hora.

3.4.6 Ajustes de la clave de API.

- ***xpack.security.auth.api_key.enabled***: evita que se pueda obtener la clave de API a través de http. El valor predeterminado es true a menos que xpack.security.http.ssl.enabled
- ***xpack.security.authc.api_key.hashing.algorithm***: algoritmo de hashing pbkdf2 que es una variación del HMAC-SHA512 como función pseudoaleatoria utilizando 10000 iteraciones
- ***xpack.security.authc.api_key.cache.ttl***: El tiempo de vida de las entradas de la clave API en caché.
- ***xpack.security.authc.api_key.cache.hash_algo***: El algoritmo de hash que se utiliza para las credenciales de la clave en caché. . El valor predeterminado es ssha25 que usa el sha256 con sal.

3.4.7 Configuraciones de auditoría

- ***xpack.security.audit.enabled***: sirve para habilitar la auditoría de logs en el nodo. El valor por defecto es false.
- ***xpack.security.audit.outputs***: especifica dónde se imprimen los registros de auditoría. Por ejemplo: [index, logfile]. El valor por defecto es logfile, que pone los eventos de auditoría en un archivo dedicado llamado <clustername>_audit.log en cada nodo.

3.4.8 Eventos de auditoría

Los eventos y alguna otra información sobre lo que se registra pueden ser controlados usando los siguientes ajustes:

- ***xpack.security.audit.logfile.events.include***: especifica qué eventos se incluyen en la salida de los logs.
- ***xpack.security.audit.logfile.events.exclude***: excluye los eventos especificados de la salida. Por defecto, no se excluye ningún evento.
- ***xpack.security.audit.logfile.events.emit_request_body***: especifica si se incluye el los datos de la solicitud de las peticiones REST en ciertos tipos de eventos como pueda ser la autenticación. El valor por defecto es false.

3.5 Ficheros de seguridad

Los ficheros de seguridad son los ficheros que usa Elastic para definir diferentes características en el servicio, estos archivos no están cifrados y están en texto plano para poder modificarse al gusto del usuario.

Todos los archivos que utilizan las funciones de seguridad deben almacenarse en el directorio de configuración de Elasticsearch. Elasticsearch se ejecuta con permisos restringidos y sólo se le permite leer desde las ubicaciones configuradas en la disposición del directorio para mejorar la seguridad. [8]

Los archivos son los siguientes:

- *<ruta_configuracion_Elastic>/roles.yml*: define los roles en uso en el cluster.
- *<ruta_configuracion_Elastic>/elasticsearch-users*: define los usuarios y sus contraseñas hash para el dominio de archivos.
- *<ruta_configuracion_Elastic>/elasticsearch-users_role*s: define la asignación de roles de usuario para el dominio de archivos.
- *<ruta_configuracion_Elastic>/role_mapping.yml*: define las asignaciones de roles para un Nombre Distinguido (DN) a un rol. Esto permite asignar grupos y usuarios de LDAP y Active Directory y usuarios PKI a los roles.
- *<ruta_configuracion_Elastic>/log4j2.properties*: contiene información de auditoría.

- *<ruta_configuracion_Elastic>elasticsearch.yml* para la configuración de los apartados de seguridad

4. Configuración de TLS, roles y autenticación en elastic.

Seguridad en la capa de transporte o *Transport Layer Security* (TLS) es el nombre de un protocolo estándar de la industria para aplicar controles de seguridad (como el cifrado) a las comunicaciones de red. TLS es el nombre moderno de lo que antes se llamaba Secure Sockets Layer (SSL).

El Protocolo de Transporte o *Transport Protocol* es el nombre del protocolo que los nodos de Elasticsearch utilizan para comunicarse entre sí. Este nombre es específico de Elasticsearch y distingue el puerto de transporte (por defecto 9300) del puerto HTTP (por defecto 9200). Los nodos se comunican entre sí utilizando el puerto de transporte, y los clientes REST se comunican con Elasticsearch utilizando el puerto HTTP.

Aunque la palabra transporte aparece en ambos protocolos, no significan lo mismo. Es posible aplicar TLS tanto al puerto de transporte de Elasticsearch como al puerto HTTP. [6]

Los nodos de Elasticsearch almacenan datos que pueden ser confidenciales. Los ataques a los datos pueden provenir de la red. Estos ataques podrían incluir el rastreo de los datos, la manipulación de los datos y los intentos de acceder al servidor y, por tanto, a los archivos que almacenan los datos. Es necesario asegurar los nodos para poder garantizar la integridad y autenticidad de los datos, por tanto, se deberá proteger la comunicación entre nodos así como la comunicación con los clientes.

Para encriptar el tráfico hacia, desde y dentro de un cluster de Elasticsearch usando SSL/TLS, requiere que los nodos se autentifiquen al unirse al clúster usando certificados SSL, haciendo más difícil para los atacantes remotos emitir cualquier comando a Elasticsearch.

La autenticación de los nuevos nodos ayuda a evitar que un nodo deshonesto se una al clúster y reciba datos a través de la replicación. [7]

Para poder usar las funciones de encriptación TLS/SSL se debe de tener en cuenta que los clusters que no tienen la encriptación habilitada por defecto y envían todos los datos en texto plano, incluyendo las contraseñas.

4.1 Roles y usuarios de elasticsearch

Por defecto Elastic configura un dominio nativo. Un dominio nativo es un dominio interno donde los usuarios se almacenan en un índice dedicado de Elasticsearch. Este dominio admite un token de autenticación en forma de nombre de usuario y contraseña, y está disponible por defecto cuando no se configura explícitamente ningún dominio. Los usuarios se gestionan a través de las API de gestión de usuarios.

La autenticación identifica a un individuo. Para obtener acceso a los recursos restringidos, un usuario debe demostrar su identidad, a través de contraseñas, credenciales, o algún otro medio (normalmente referido como tokens de autenticación).

Elastic autentifica a los usuarios identificando a los usuarios detrás de las solicitudes que llegan al clúster y verificando que son quienes dicen ser. El proceso de autenticación es manejado por uno o más servicios de autenticación llamados *realm*.

Se puede utilizar el soporte nativo para gestionar y autenticar usuarios, o integrar sistemas de gestión de usuarios externos como LDAP y Active Directory.

Las características de seguridad de Elastic proveen dominios (realm) integrados como nativo, ldap, active_directory, pki, file, y saml. También se puede construir tu propio dominio (realm) personalizado y conectarlo a Elastic.

Cuando las características de seguridad están habilitadas, dependiendo de los dominios que se hayan configurado, se debe adjuntar las credenciales de usuario a las solicitudes enviadas a Elastic. Por ejemplo, cuando se utilizan

dominios que admiten nombres de usuario y contraseñas, puede simplemente adjuntar el encabezado de autenticación básica a las solicitudes.

Las características de seguridad proporcionan dos servicios: el servicio de token y el servicio de clave api. Se puede utilizar estos servicios para intercambiar la autenticación actual por un token o una clave. Este token o clave puede utilizarse como credencial para autenticar nuevas solicitudes. Estos servicios están habilitados por defecto cuando se activa TLS/SSL para HTTP.

Las funciones de seguridad de Elastic proporcionan credenciales de usuario integradas para ayudar a la puesta en marcha. Estos usuarios tienen un conjunto fijo de privilegios y no pueden ser autenticados hasta que sus contraseñas hayan sido establecidas.

- *elastic:* Es el nombre del usuario con permisos de administración total en el servicio.
- *kibana:* El usuario que utiliza Kibana para conectarse y comunicarse con Elasticsearch.
- *logstash_system:* El usuario que usa Logstash cuando almacena información de monitoreo en Elastic.
- *beats_system:* El usuario que utiliza Beats cuando almacena la información de monitorización en Elastic.
- *apm_system:* El usuario que utiliza el servidor de APM al almacenar la información de monitorización en Elastic.
- *remote_monitoring_user:* El usuario que Metricbeat utiliza cuando recoge y almacena información de monitorización en Elasticsearch.

Hasta la version 7.X elastic mantenía contraseñas por defectos, la cual era simplemente “changeme”, esta contraseña se podía modificar lanzando un script de Elastic dentro de los archivos de configuración.

A partir de la versión 7.x elasticsearch, permitirá acceso a los usuarios una vez se ejecuté la configuración mínima de seguridad. En esta se creará un usuario “elastic” de acceso al cluster y nodos. Existe dos métodos para generar la contraseña a este usuario con el comando

```
sudo ./elasticsearch-setup-passwords
```

1. Usando el parámetro *auto* que generará una contraseña aleatoria.
2. Usando el parámetro *interactive* el cual el usuario creará sus propias contraseñas

Una vez se ejecute uno de los dos comandos no se podrán volver a ejecutar.

Tras ejecutar el comando Elastic pedirá al usuario que contraseñas usar para cada uno de los usuarios.

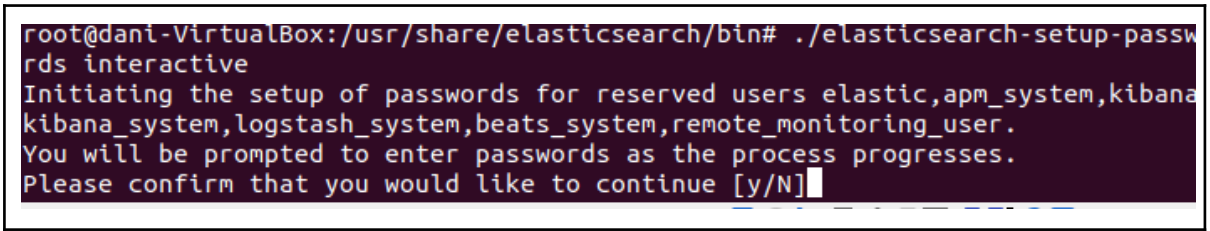


Figura 5: Configuración de contraseñas.

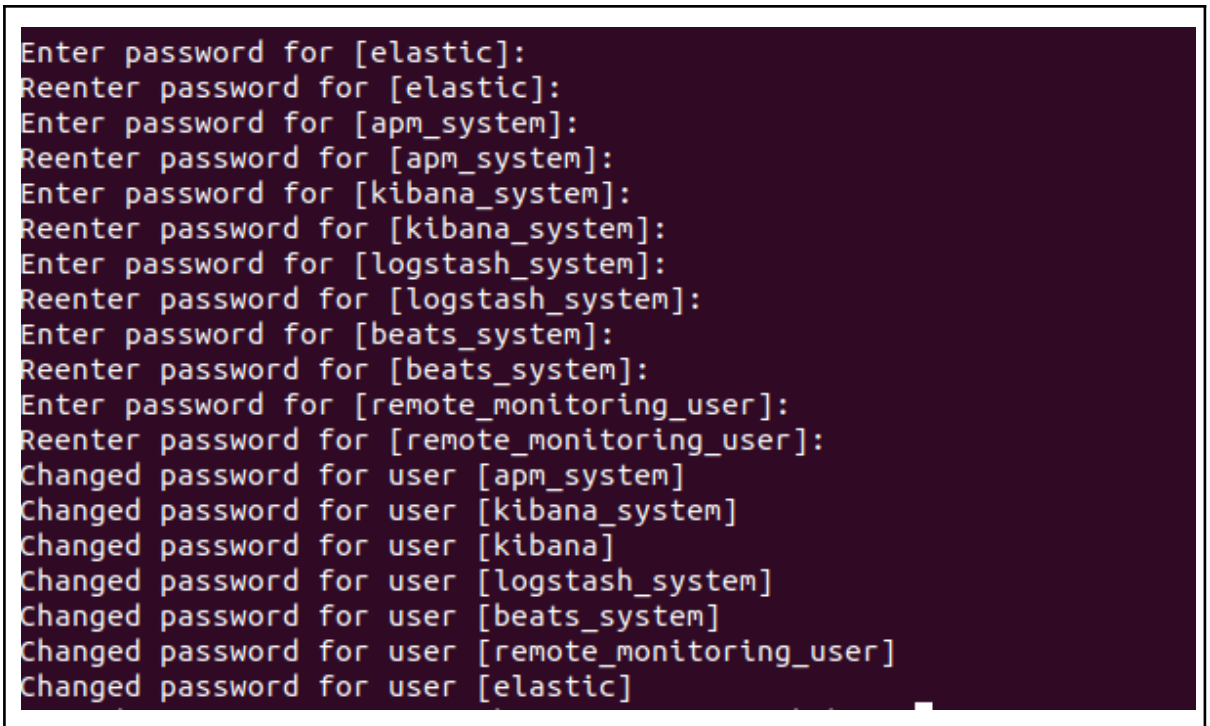


Figura 6: Creación de usuarios y contraseñas.

Una vez hecho esto, cuando se intente hacer una petición a elastic habrá que autentificarse con alguna cuenta de usuario de la imagen anterior.

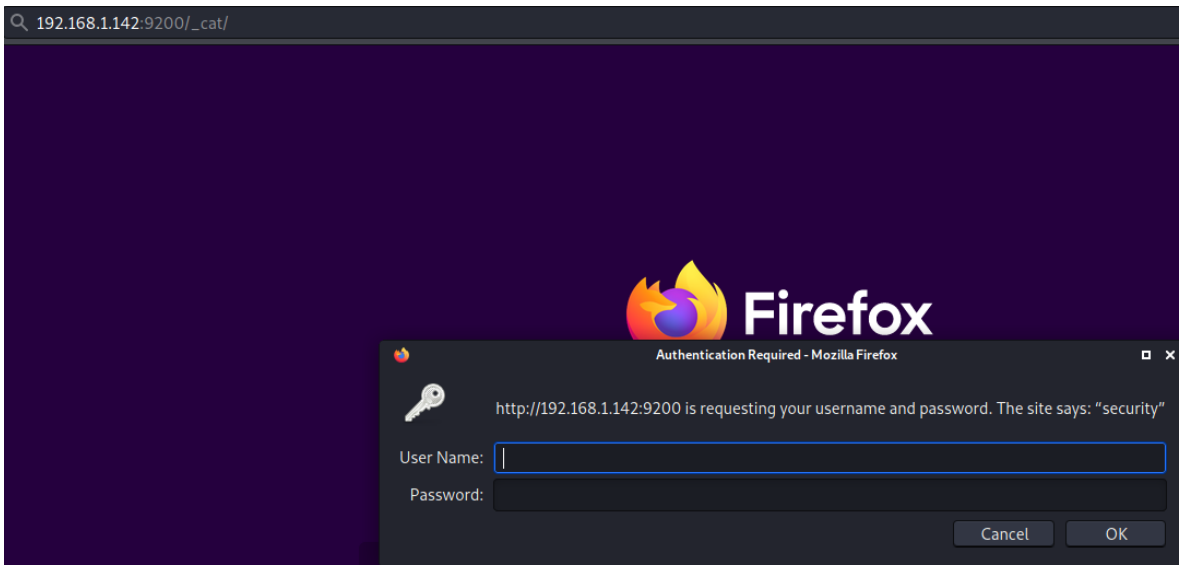


Figura 7: Petición a un servicio Elastic con usuarios.

En caso de no escribir ningún usuario o escribir una contraseña o usuario erróneo se lanzará la siguiente página de error.

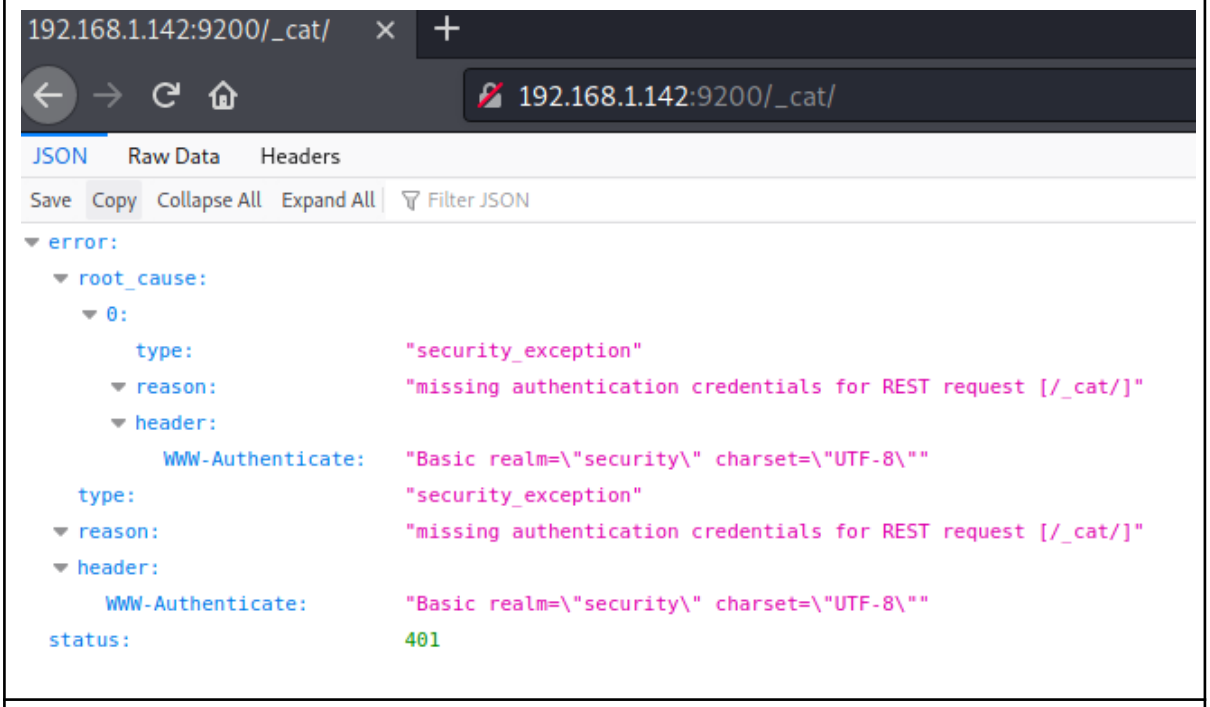


Figura 8: Elastic detecta que no ha habido autenticación.

4.2- Comunicación segura entre nodos de un cluster

El siguiente paso es añadir TLS al protocolo de transporte, el protocolo del que se ha hablado anteriormente que permite a los nodos comunicarse entre sí, con TLS, las comunicaciones entre los nodos estarán cifradas.

Se pueden añadir tantos nodos como se quiera en un clúster, pero deben ser capaces de comunicarse entre sí. La comunicación entre los nodos de un clúster la gestiona el módulo de transporte. Para asegurar un cluster, hay que asegurarse de que las comunicaciones entre nodos están encriptadas y verificadas, lo cual se logra con el protocolo TLS.

En un clúster seguro, los nodos de Elastic utilizan certificados para identificarse cuando se comunican con otros nodos.

El cluster debe validar la autenticidad de estos certificados. El enfoque recomendado es confiar en una autoridad de certificación[1] (CA) específica. Cuando

[1] Una autoridad de certificación es una entidad de confianza responsable de emitir y revocar los certificados digitales

los nodos se añaden a su clúster, deben utilizar un certificado firmado por la misma CA.

Para configurar el TLS se podrá utilizar la herramienta elasticsearch-certutil que generará una CA para el clúster:

```
/usr/share/elasticsearch/bin/elasticsearch-certuil CA
```

Figura 9: Generación de una CA.

A continuación se deberá generar un clave privada para cada nodo usando la CA generada anteriormente.

```
./usr/share/elasticsearch/bin/elasticsearch-certutil cert --ca
<nombre_certificad>
```

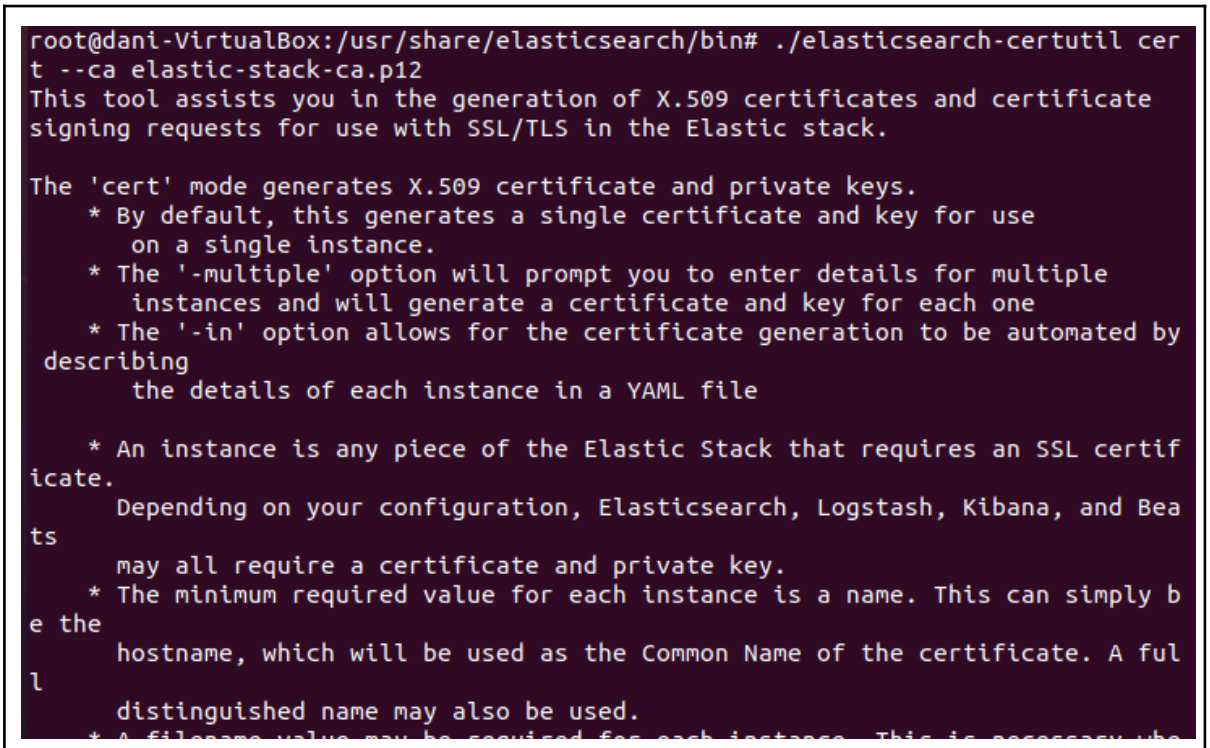


Figura 10: Generación del certificado autofirmado

Una vez realizado estos pasos los archivos generados en la carpeta de configuración de elasticsearch. Posteriormente habrá que escribir las siguientes líneas en el archivo de configuración elastichsearch.yml:

```
xpack.security.transport.ssl.verification_mode: certificate
xpack.security.transport.ssl.client_authentication: required
xpack.security.transport.ssl.trustore.path: <nombre_certificado>
```

4.3- Comunicaciones seguras en el protocolo de transporte en elasticsearch

Para proteger el protocolo de transporte y se mantengan todas las comunicaciones seguras entre los clientes y el servidor Elastic habrá que añadir una capa más de seguridad, protegiendo así la integridad de los datos.

Esta última capa es sencilla de configurar, tan solo habrá que moverse a la carpeta donde se han ejecutado anteriormente los scripts que Elastic trae y ejecutar la siguiente orden.

```
./bin/elasticsearch-certutil http
```

Este script genera un certificado y la clave, habrá que indicar la ruta de la ca que se generó en el apartado anterior.

Por último, en el archivo de configuración añadiremos las siguientes líneas:

```
xpack.security.transport.ssl.enabled: true
xpack.security.transport.ssl.keystore.path: <nombre_certificado>
```

5.- Instalación de elasticsearch en ubuntu

En este capítulo se va a exponer el proceso de instalación usado para la instalación de Elastic

Todos los paquetes de Elastic están firmados con la clave de firma de Elasticsearch para proteger su sistema contra la suplantación de paquetes. El administrador de paquetes considerará confiables los paquetes autenticados con la clave. En este paso, se importará la clave GPG pública de Elasticsearch y se agregará a la lista de fuentes de paquetes de Elastic para instalar Elasticsearch en la versión 7.14.1. [9]

```
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch |
sudo apt-key add -
```

Además habrá que tener instalado el paquete apt-transport-https en Debian antes de continuar:

```
sudo apt-get install apt-transport-https
```

También se deberá añadir la lista de fuentes de Elastic al directorio sources.list.d, donde APT buscará nuevas fuentes:

```
echo "deb https://artifacts.elastic.co/packages/7.x/apt stable
main" | sudo tee /etc/apt/sources.list.d/elastic-7.x.list
```

Por último, se deberá actualizar la lista de paquetes e instalar elasticsearch

```
sudo apt-get update && sudo apt-get install elasticsearch
```

Para configurar Elasticsearch, se editará su archivo de configuración principal elasticsearch.yml, donde se almacena la mayoría de sus opciones de configuración. Este archivo se encuentra en el directorio /etc/elasticsearch

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

El archivo elasticsearch.yml ofrece opciones de configuración para su clúster, nodo, rutas, memoria, red, detección y puerta de enlace. La mayoría de estas opciones están preconfiguradas en el archivo, pero se puede cambiar según las necesidades.

Elasticsearch escucha el tráfico de todos los lugares en el puerto 9200. Es conveniente restringir el acceso externo a la instancia de Elasticsearch para evitar que terceros lean los datos o cierren su clúster de Elasticsearch a través de su API REST. Para restringir el acceso y, por lo tanto, aumentar la seguridad, habrá que buscar la línea que especifica network.host, eliminar los comentarios y reemplazaar su valor por localhost para que diga:

```
/etc/elasticsearch/elasticsearch.yml
network.host: localhost
```

Al especificar localhost, Elasticsearch escuchará en todas las interfaces y las IP vinculadas. También se puede indicar que escuche únicamente en una interfaz específica, se puede especificar una IP en lugar de localhost. Ahora habrá que guardar y cerrar elasticsearch.yml.

Estos son los ajustes mínimos con los que puede comenzar para usar Elasticsearch. Ahora, se puede iniciar Elasticsearch por primera vez después de ejecutar los siguientes comandos para permitir que Elasticsearch se cargue cada vez que su servidor se inicie

```
sudo systemctl enable elasticsearch
sudo systemctl start elastichsearch
```

Para acceder a elastic se usará la siguiente url: http://localhost:9200

6.- Pentesting a Elasticsearch

En este capítulo se encuentra la realización del pentesting hecho a la servicio aislado que se ha preparado, se parte de la base en que el servicio Elastic esta con la configuración por defecto al instalar este servicio.

6.1 Puesta en marcha de las máquinas a utilizar

Para la realización de este pentesting se han usado 2 máquinas virtuales:

1.- La primera contiene un sistema operativo Ubuntu 20.04 al cual se le ha instalado la última versión disponible de elasticsearch y hará de servidor y víctima. Este servicio de elasticsearch ha sido rellenado con datos "*dummy*" para poder tener lo más cercano a un sistema real. IP de la máquina: 192.168.1.141

2.- La segunda es la máquina será la que atacará a la víctima, la cual ejecuta un sistema operativo Kali Linux en su última versión. IP de la máquina 192.168.1.140

6.2 Hacking ético

El hacking ético es una forma de referirse al acto de una persona, que utiliza sus conocimientos de informática y seguridad para encontrar vulnerabilidades o fallas de seguridad en el sistema, una vez hecho esto serán reportadas a la organización pertinente para que precises los cambios convenientes para la mejora de la seguridad.

El protocolo usado por Elasticsearch es el HTTP como se ha explicado anteriormente. Cuando se accede a este via HTTP, se podrá encontrar información interesante sobre el sistema donde está corriendo el servicio con la siguiente petición GET http://IpDestino:9200/, para este caso http://192.168.141.1:9200/

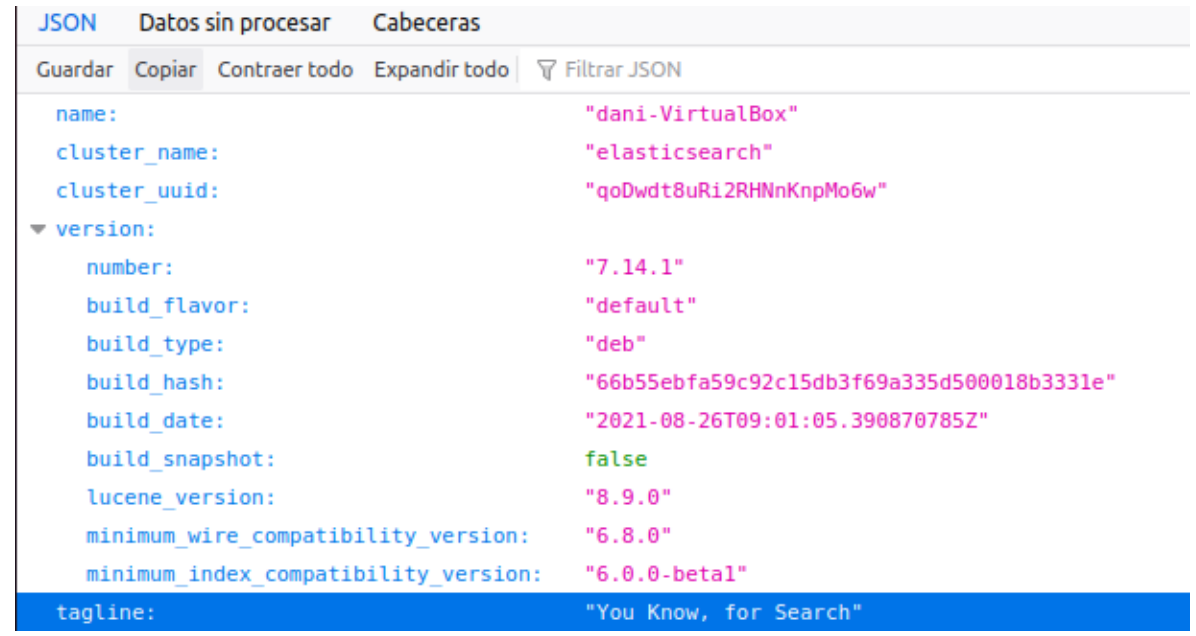


Figura 11: Banner Grabbing.

Por defecto, como en Elasticsearch no está activada la autenticación, se puede conseguir acceso a toda la base de datos sin necesidad de ninguna credencial.

Para comprobar si se tiene habilitado la autenticación en Elasticsearch tan solo habrá que hacer uso del comando curl de la siguiente manera:

```
curl -X GET 192.168.1.141:9200/_xpack/security/user
```

Lo que devolverá la siguiente respuesta en formato JSON

```
{
  "error": {
    "root_cause": [
      {
        "type": "exception",
        "reason": "Security must be explicitly enabled when using a [basic] license. Enable security by
setting [xpack.security.enabled] to [true] in the elasticsearch.yml file and restart the node."
      }
    ],
    "type": "exception",
    "reason": "Security must be explicitly enabled when using a [basic] license. Enable security by setting
[xpack.security.enabled] to [true] in the elasticsearch.yml file and restart the node."
  },
  "status": 500
}
```


Figura 12: Comprobar si existen usuarios en el servicio.

Esta respuesta indica que no está activada la autenticación. De hecho si la petición se hace directamente a 192.168.1.141:9200/_xpack, se puede observar que en la respuesta, la autenticación está disponible pero no está habilitada.

```
curl -X GET 192.168.1.141:9200/_xpack/
```


```
{"build":
{"hash":"66b55ebfa59c92c15db3f69a335d500018b3331e","date":"2021-08-26T09:01:05.390870785Z"},"license":
{"uid":"af26729a-a0c5-4158-8023-b5c0beacf9d7","type":"basic","mode":"basic","status":"active"},"features":
{"aggregate_metric":{"available":true,"enabled":true},"analytics":{"available":true,"enabled":true},"ccr":
{"available":false,"enabled":true},"data_streams":{"available":true,"enabled":true},"data_tiers":
{"available":true,"enabled":true},"enrich":{"available":true,"enabled":true},"eql":
{"available":true,"enabled":true},"frozen_indices":{"available":true,"enabled":true},"graph":
{"available":false,"enabled":true},"ilm":{"available":true,"enabled":true},"logstash":
{"available":false,"enabled":true},"ml":{"available":false,"enabled":true,"native_code_info":
{"version":"7.14.1","build_hash":"696c4ee37e496c"}},"monitoring":
{"available":true,"enabled":true},"rollup":{"available":true,"enabled":true},"searchable_snapshots":
{"available":false,"enabled":true},"security":{"available":true,"enabled":false},"slm":
{"available":true,"enabled":true},"spatial":{"available":true,"enabled":true},"sql":
{"available":true,"enabled":true},"transform":{"available":true,"enabled":true},"vectors":
{"available":true,"enabled":true},"voting_only":{"available":true,"enabled":true},"watcher":
{"available":false,"enabled":true}},"tagline":"You know, for X"}
```


Figura 13: Comprobar que paquetes tiene habilitados el servicio

Por tanto, teniendo acceso a toda la API REST de elastic, están todas las peticiones vía GET que podemos hacer.

Tabla 1: Llamadas a la api que se pueden hacer.

/_cat	**/_cluster**	**/_security**

/_cat/segments	/_cluster/allocation/explain	/_security/user
/_cat/shards	/_cluster/settings	/_security/privilege
/_cat/repositories	/_cluster/health	/_security/role_mapping
/_cat/recovery	/_cluster/state	/_security/role
/_cat/plugins	/_cluster/stats	/_security/api_key
/_cat/pending_tasks	/_cluster/pending_tasks	
/_cat/nodes	/_nodes	
/_cat/tasks	/_nodes/usag	
/_cat/templates	/_nodes/hot_threads	
/_cat/thread_pool	/_nodes/stats	
/_cat/ml/trained_models	/_tasks	
/_cat/transforms/_all	/_remote/info	
/_cat/aliases		
/_cat/allocation		
/_cat/ml/anomaly_detectors		
/_cat/count		
/_cat/ml/data_frame/analytic		
/_cat/ml/datafeeds		
/_cat/fielddata		

/_cat/health		
/_cat/indices		
/_cat/master		
/_cat/nodeattrs		

Las llamadas a la api más interesantes son:

- /_security/user: Si se tiene habilitada la autorización, se puede comprobar que usuario es el “superusuario”
- /_cat/indices: Se puede reunir toda la información sobre los índices que contiene el servicio elastic
- /_cat/<indice>: Para obtener información sobre qué tipo de datos se guardan dentro de un índice
- /_cat/nodes: Lista los nodos activos del servicio elasticsearch

Se puede acceder a todos los índices de la plataforma elastic accediendo a http://192.168.1.141:9200/_cat/indices?v

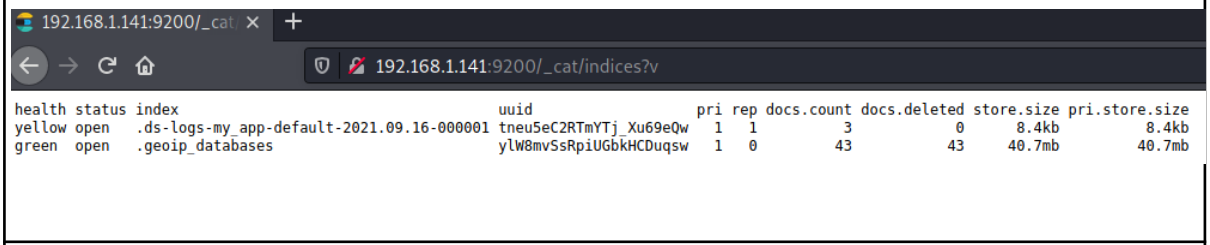


Figura 14: Índices del servicio.

Para obtener toda la información de un índice tan solo hay que mandar una petición de la siguiente manera: *http://IpDestino:9200/<Nombreindice>*. Para este

caso la petición sería *http://192.168.1.141:9200/.ds-logs-my_app-default-2021.09.16-000001*, la cual nos devolvería toda la información sobre este index.

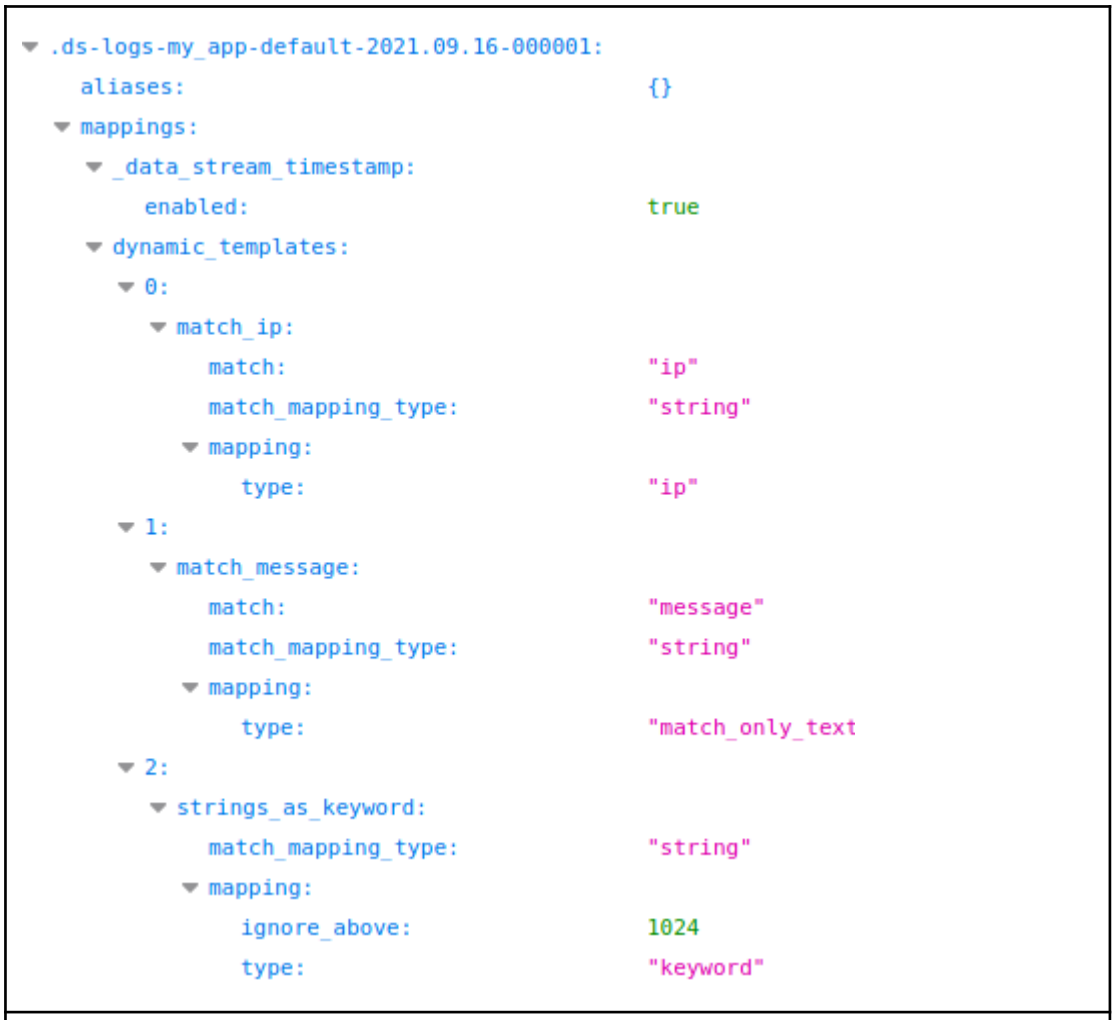
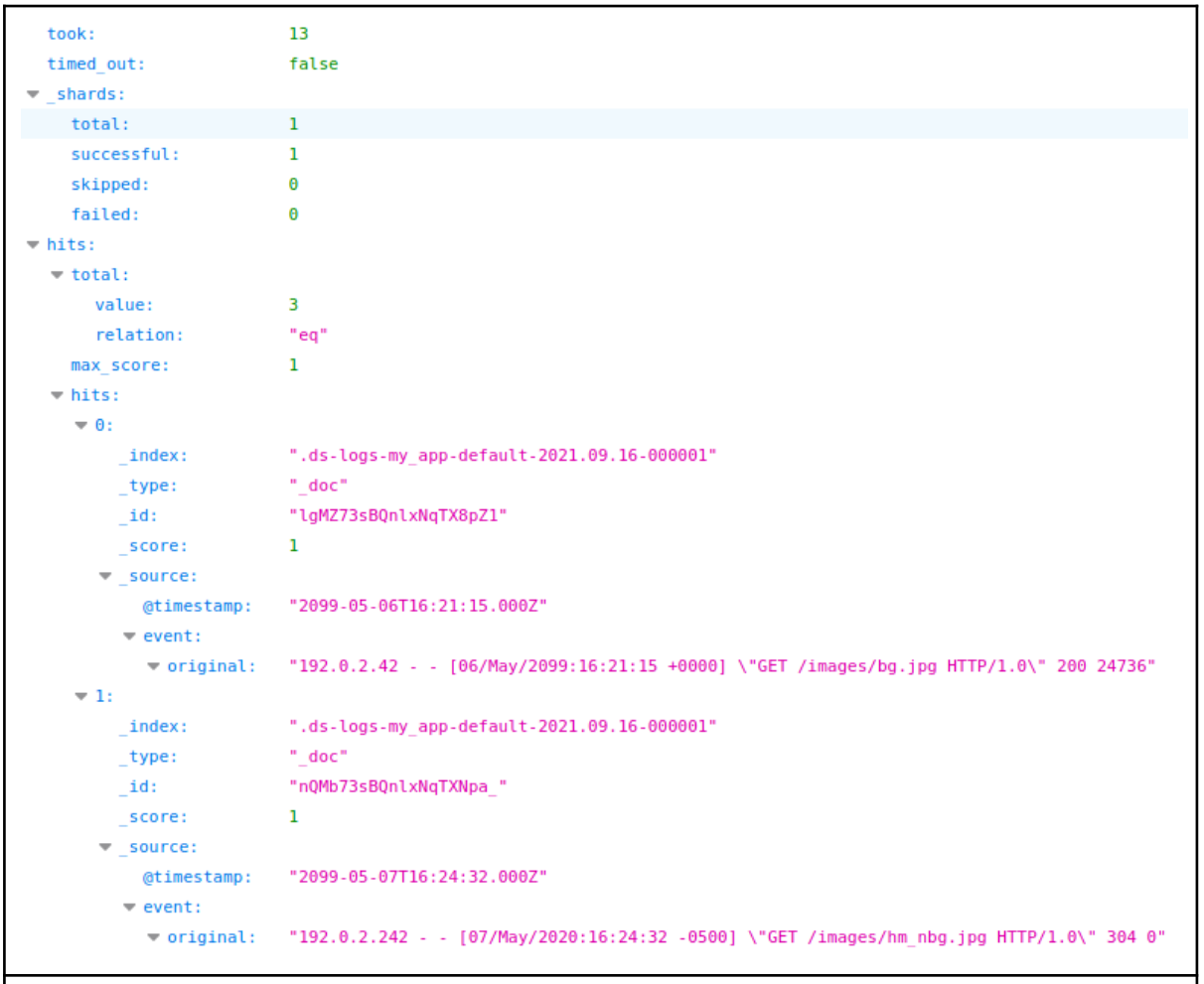


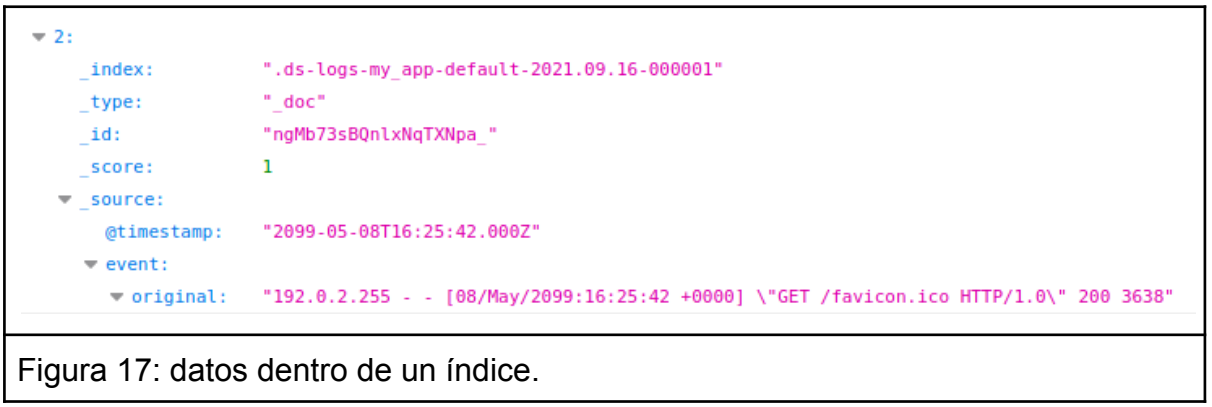
Figura 15: Tipo de datos de un índice consultado.

Si se quiere volcar todo el contenido de un índice se puede acceder a: *http://IpDestino:9200/<Nombreindice>/_search?pretty=true, para este caso, http://192.168.1.141:9200/.ds-logs-my_app-default-2021.09.16-000001/_search?pretty=true*



```
took:                 13
timed_out:            false
_shards:
  total:              1
  successful:         1
  skipped:            0
  failed:             0
hits:
  total:
    value:            3
    relation:         "eq"
  max_score:          1
  hits:
    0:
      _index:         ".ds-logs-my_app-default-2021.09.16-000001"
      _type:          "_doc"
      _id:            "lgMZ73sBQnlxNqTX8pZ1"
      _score:         1
      _source:
        @timestamp:   "2099-05-06T16:21:15.000Z"
        event:
          original:   "192.0.2.42 - - [06/May/2099:16:21:15 +0000] \"GET /images/bg.jpg HTTP/1.0\" 200 24736"
    1:
      _index:         ".ds-logs-my_app-default-2021.09.16-000001"
      _type:          "_doc"
      _id:            "nQMb73sBQnlxNqTXNpa_"
      _score:         1
      _source:
        @timestamp:   "2099-05-07T16:24:32.000Z"
        event:
          original:   "192.0.2.242 - - [07/May/2020:16:24:32 -0500] \"GET /images/hm_nbg.jpg HTTP/1.0\" 304 0"
```


Figura 16: Datos dentro de un índice.



```
2:
  _index:         ".ds-logs-my_app-default-2021.09.16-000001"
  _type:          "_doc"
  _id:            "ngMb73sBQnlxNqTXNpa_"
  _score:         1
  _source:
    @timestamp:   "2099-05-08T16:25:42.000Z"
    event:
      original:   "192.0.2.255 - - [08/May/2099:16:25:42 +0000] \"GET /favicon.ico HTTP/1.0\" 200 3638"
```


Figura 17: datos dentro de un índice.

Como se puede observar, la primera de las peticiones al índice, indica que tipo de datos existen dentro de este, y la segunda, muestra directamente los datos

que hay en el mismo.

Para volcar todo tan solo hay que usar la misma ruta que antes pero sin indicar ningún índice http://IPDestino:9200/_search?pretty=true, para este caso, http://192.168.1.141:9200/_search?pretty=true. Aunque solo se tiene un límite de 10 resultados por defecto, se puede utilizar el parámetro *size* para volcar tantos resultados como se desee.

Si está buscando alguna información en específico, se pueden hacer búsquedas en bruto en todos los índices que van a http://IPDestino:9200/_search?pretty=true&q=<término_de_búsqueda>, en este caso, http://192.168.1.141:9200/_search?pretty=true&q=cd. Aunque también se puede buscar dentro de un índice en específico añadiendo el índice de la siguiente manera: http://192.168.1.141:9200/nuevoslogs/_search?pretty=true&q=cd.

Figura 18: Búsqueda de datos específicos mediante “search”.

Para comprobar los permisos de escritura que se tiene tan solo hay probar a

crear un nuevo índice con un documento utilizando el comando curl:

```
curl -X POST '192.168.1.141:9200/escribirindice/escribirdoc' -H
'Content-Type: application/json' -d'
{
"PermisoEscritura" : "Lo tengo"
}'
```

Si ahora se hace una petición para ver todos los índices a http://192.168.1.141:9200/_cat/indices se puede observar el nuevo indice creado en caso de tener permisos:

```
green  open .geoip_databases                       ylW8mvSsRpiUGbkHCDuqsw 1 0 43 43 40.7mb 40.7mb
yellow open .ds-logs-my_app-default-2021.09.16-000001 tneu5eC2RTmYTj_Xu69eQw 1 1  3  0  8.4kb  8.4kb
yellow open nuevoslogs                             j9RKKZ60RlW3W7bXZHubdw 1 1  3  0 16.9kb 16.9kb
yellow open escribirindice                         R4BZMiyDTwOm-RnCaaiyGQ 1 1  1  0  3.8kb  3.8kb
```

Figura 19: Listado de índices tras la creación de uno mediante API REST.

Si ahora se activa el paquete xpack.security, activamos la autenticación, por tanto al hacer curl se puede observar que las peticiones son fallidas debido a que es necesario la autenticación del usuario.

```
curl -X GET 192.168.1.141:9200/_xpack/security/user
```

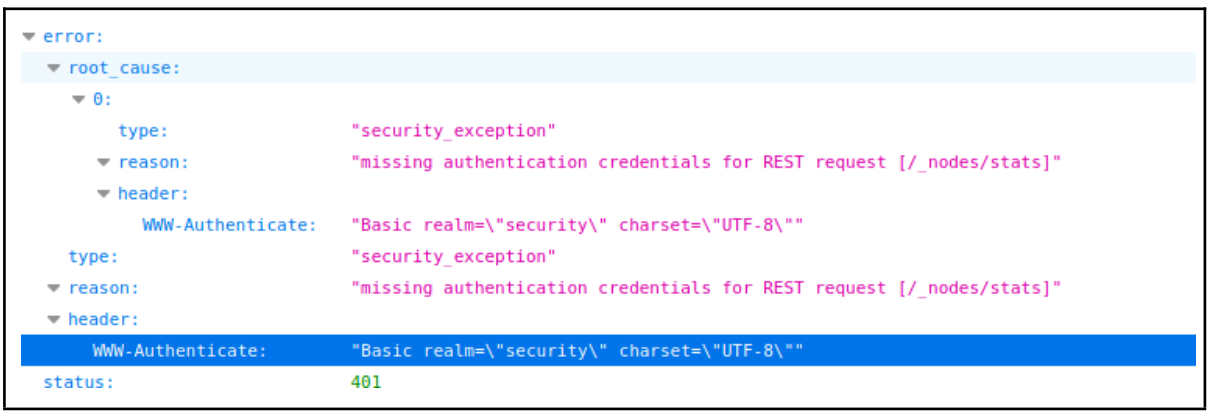


Figura 20: Petición fallida debido a que es necesario autenticación.

6.3 Enumeración

Aunque se ha visto en este documento la facilidad con la que se puede acceder, robar e incluso modificar los datos si se deja la configuración por defecto en Elastic, existen una cantidad ingente de servicios Elastic con esta configuración expuestos a internet.

Para poder encontrar estos servicios sin configuración alguna, se ha utilizado la herramienta Shodan. Shodan es un motor de búsqueda que le permite al usuario encontrar iguales o diferentes tipos específicos de equipos (routers, servidores, etc.) conectados a Internet a través de una variedad de filtros. Algunos también lo han descrito como un motor de búsqueda de banners de servicios [10].

Para este caso tan solo han sido necesarios 2 filtros:

- *port*: indica sobre qué puerto desea que se realice la búsqueda de banners.
- Palabra clave: por ejemplo, el nombre del servicio que se está buscando, si este se encuentra en el banner, lo devolverá como resultado.

Por tanto, el filtro que se ha utilizado es:

```
port: 9200 elasticsearch
```

Como resultado de esta búsqueda se han encontrado un total de 3870 resultados como se puede ver en la siguiente imagen.

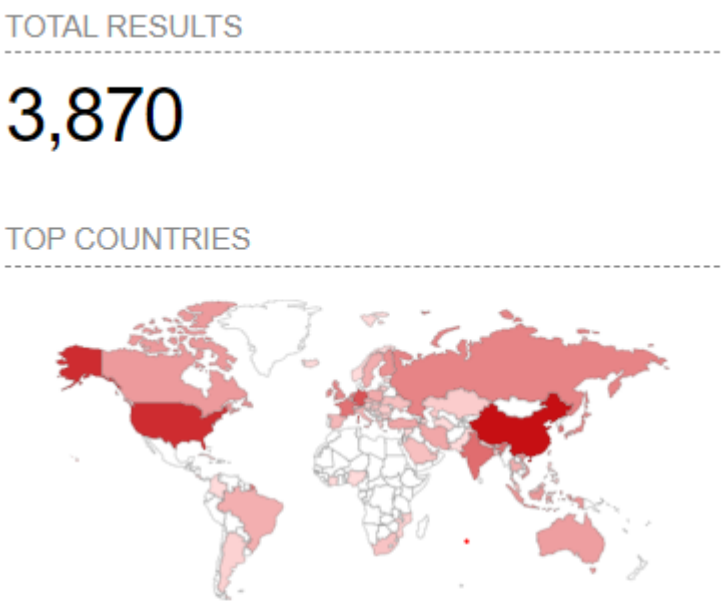


Figura 21: Número de servicios escaneados con Shodan.

Ahora bien, echando un simple vistazo se puede notar que en la mayoría de los banners devuelven que las características de seguridad no están habilitadas, lo que lleva a poder usar todas las técnicas que se han visto en el apartado anterior, la siguiente imagen muestra las ips expuestas

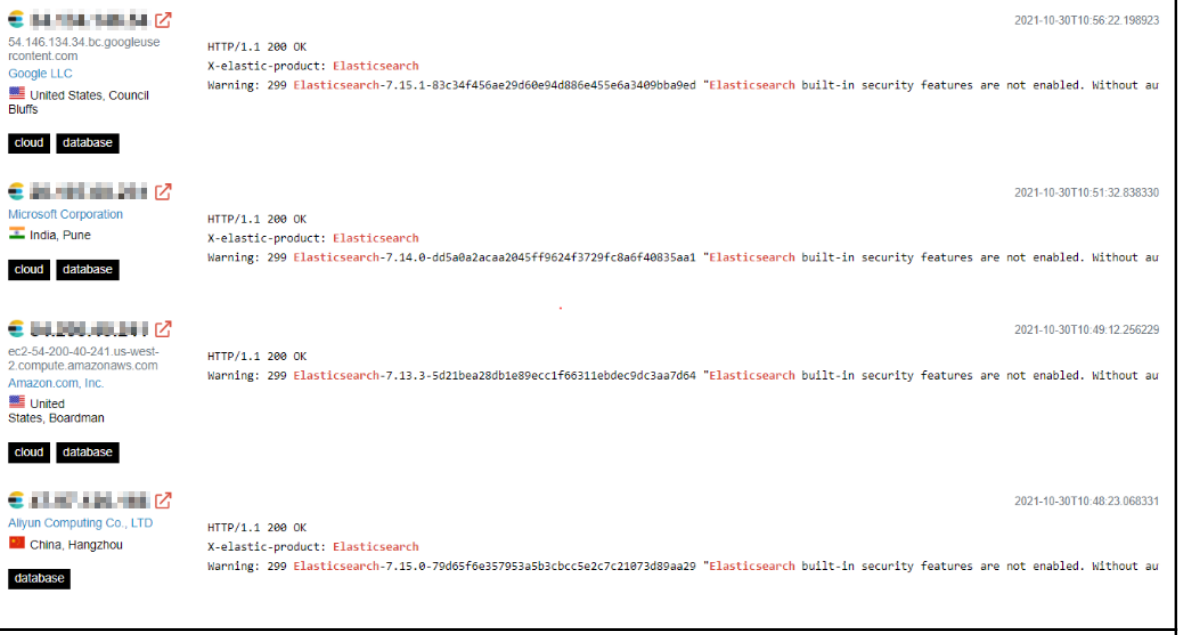


Figura 22: Banners de servicios escaneados por Shodan.

A modo de exponer aquí lo que se ha explicado en el apartado anterior y de forma no intrusiva, se ha accedido a los servicios elasticsearch de las ips encontradas para comprobar que realmente son servicios que realmente contienen datos sensibles y que podrían ser fácilmente manipulados y extraídos sin necesidad de explotar ninguna vulnerabilidad ya que no tienen habilitado ningún tipo de configuración de seguridad.

En la siguiente imagen se puede observar como el servicio Nextcloud está totalmente expuesto y accesible, el cual contiene 8Gb de datos.

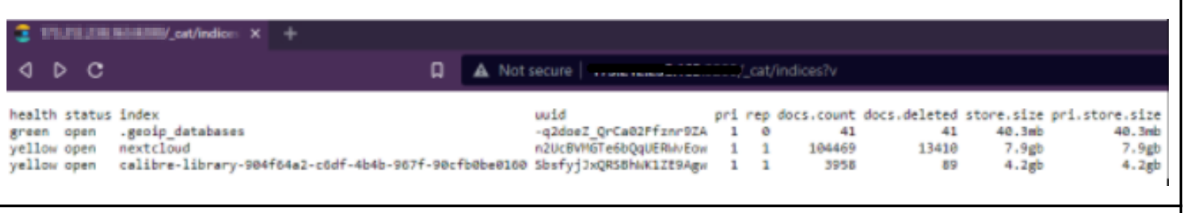


Figura 23: Listado de índices de un servicio expuesto a internet.

Nextcloud, es una herramienta completa orientada a empresas y particulares, cuya función es actuar como un servidor de almacenamiento en la nube de fotos, datos, archivos. Está desarrollado con el objetivo de ofrecer rentabilidad y productividad a empresas, aunque su uso es completamente útil para particulares. Por ello permite una gran personalización a través de la instalación de apps o módulos, que permite ampliar funcionalidades más completas, según las necesidades. [11]

Si entramos dentro del índice podemos ver que realmente existen archivos con contenido, por ejemplo en esta imagen.

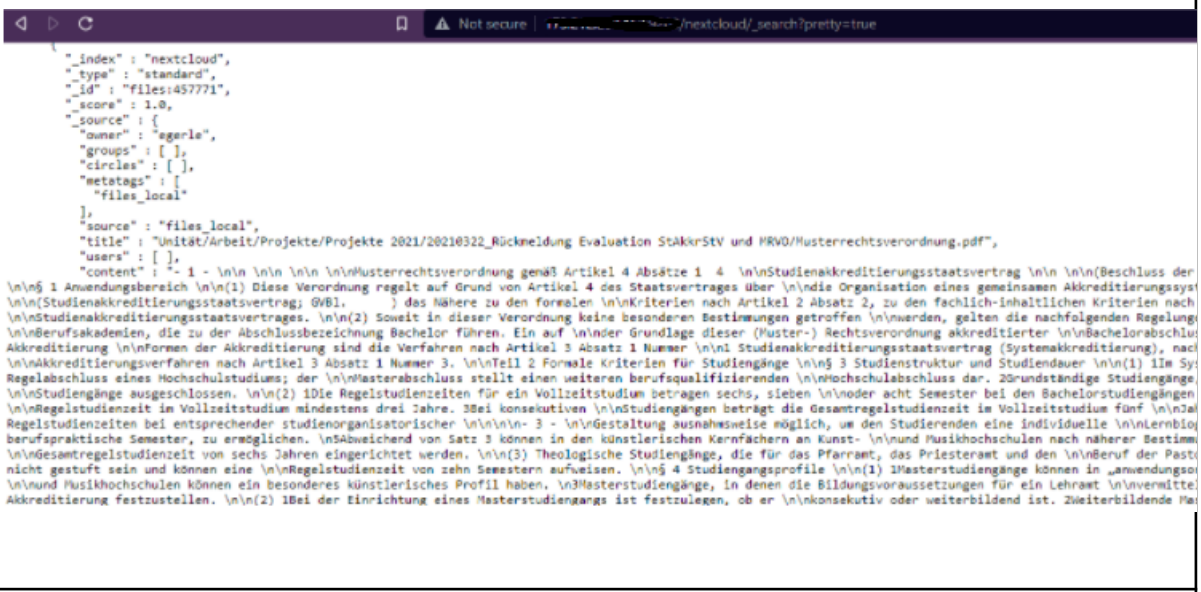


Figura 24: Acceso a un documento interno del servicio expuesto

6.4 Análisis de riesgos

Tabla 2: Exposición del servicio

DESCRIPCIÓN	RIESGO
Exposición de información sensible a un agente no autorizado	Alto
Inyección de datos desde un agente no autorizado	Alto
Denegación de servicio de un nodo sin autorización	Medio
No hay encriptación en la comunicación cliente-servidor	Alto
Comunicación no encriptada entre los nodos de un cluster	Medio

El no tener habilitado ninguna configuración de seguridad causa un gran riesgo de robos de información, integridad de los datos, fugas de información no deseadas, denegación de nodos, etc.

7.- Vulnerabilidades y ataques a Elasticsearch

En este apartado se expone las diferentes vulnerabilidades que sufre Elasticsearch, se encontrará un listado con los CVE y las versiones que sufren dicha vulnerabilidad.

7.1.- Vulnerabilidades 7.x

Aún habiendo expuesto la facilidad con la que se puede vulnerar una plataforma Elastic si no se realizan configuraciones de seguridad, todavía se puede seguir vulnerando la plataforma por muchas configuraciones de seguridad que se tengan, ya que siguen existiendo brechas de seguridad como ocurren con los siguientes CVEs.

- CVE-2021-22132: Las versiones 7.7.0 a 7.10.1 de Elasticsearch contienen un fallo de divulgación de información en la API de búsqueda asíncrona. Los usuarios que ejecutan una búsqueda asíncrona almacenan incorrectamente las cabeceras HTTP. Un usuario de Elasticsearch con la capacidad de leer el índice .tasks podría obtener cabeceras de solicitud sensibles de otros usuarios en el clúster. Este problema se ha solucionado en Elasticsearch 7.10.2.
- CVE-2021-22135: Elasticsearch versiones anteriores a 7.11.2 y 6.8.15, contienen un fallo en la divulgación de documentos que se encontró en la API suggester y profile de Elasticsearch cuando Document and Field Level Security está habilitada. La API suggester y profile normalmente están deshabilitadas para un índice cuando la seguridad a nivel de documento está habilitada en el índice. Determinadas consultas son capaces de habilitar el profiler y suggester, lo que podría conllevar a revelar la existencia de documentos y campos que el atacante no sería capaz de visualizar
- CVE-2021-22138: En Logstash versiones posteriores a 6.4.0 y anteriores a 6.8.15 y 7.12.0, se encontró un fallo de comprobación del certificado TLS en la funcionalidad de monitoreo. Cuando se especifica un certificado CA en un servidor confiable, Logstash no verificaba apropiadamente el certificado devuelto por el servidor de monitoreo. Esto podría resultar en un ataque de

tipo man in the middle contra los datos de monitoreo de Logstash.

- CVE-2020-7017: En las versiones de Kibana anteriores a la 6.8.11 y la 7.8.1 la visualización del mapa de regiones en contiene un fallo de XSS almacenado. Un atacante que sea capaz de editar o crear una visualización de mapa de regiones podría obtener información sensible o realizar acciones destructivas en nombre de los usuarios de Kibana que vean la visualización de mapa de regiones.
- CVE-2020-7016: Las versiones de Kibana anteriores a la 6.8.11 y la 7.8.1 contienen un fallo de denegación de servicio (DoS) en Timelion. Un atacante puede construir una URL que, al ser vista por un usuario de Kibana, puede hacer que el proceso de Kibana consuma grandes cantidades de CPU y deje de responder.

7.2 Vulnerabilidades 8.x

- CVE-2022-23709: las versiones afectadas son desde la 7.7.0 a la 7.17 y 8.0. Un usuario con permisos de lectura a la función de tiempo de actividad puede crear o modificar reglas de alertas, aunque estas reglas no serán habilitadas.
- CVE-2022-23710: las versiones afectadas son 7.15, 7.17 y 8.0. Se puede inyectar código malicioso en el apartado “*Data Preview Pane*”.
- CVE-2022-23712: las versiones afectadas son desde la 8.0 hasta 8.2. Un atacante no autenticado puede tirar abajo los nodos usando una cadena específica durante una petición.

Puesto que Elasticsearch versión tan solo lleva 9 meses en producción, no se han encontrado aún vulnerabilidades significativas que afecten a las últimas versiones de Elasticsearch

8.- Mejoras en la versión 8.x

En este capítulo se expondrá las distintas mejoras de Elasticsearch en su nueva versión mejorada, está versión salió a la luz durante el transcurso del proyecto (Febrero 2022) por lo que aquí se planteará si ha habido mejoras en la seguridad respecto a versión anterior 7.x. Todo lo expuesto aquí se ha realizado usando la propia documentación de Elasticsearch para esta versión [13].

8.1.- Configuración inicial

Para empezar, a partir de esta nueva versión, nada más ejecutas por primera vez un entorno de Elasticsearch, la variable ***xpack.security.enabled*** viene habilitado por defecto, algo que no ocurre en ninguna versión anterior, esto es un gran paso puesto que si esta variable no está habilitada, no se puede realizar ninguna configuración de seguridad.

Por otro lado, ya no hay que crear un usuario manualmente y crearle una contraseña, simplemente usando el comando que se muestra a continuación, se podrá cambiar la contraseña a una específica para el usuario “elastic”:

```
./bin/elasticsearch-reset-password -i -u elastic
```

Indicar que también se puede configurar la contraseña para cualquier usuario por defecto usando el mismo comando.

8.2.- Comunicaciones entre clusters

Elasticsearch, a partir de esta versión también configura de forma automática la comunicación cifrada entre los clusters. En caso de que se quiera configurar de forma manual, se tendrá que realizar esta configuración antes de iniciar los nodos, en este caso, Elasticsearch respeta la configuración y no realiza la configuración automática. De todas formas, siempre puedes modificar la configuración una vez Elasticsearch ha hecho la configuración inicial.

8.3.- Resto de apartados de seguridad

Una última mejora en la configuración de seguridad por parte de Elasticsearch ha sido que ya no admite comunicaciones hacía los clusters de forma anónima, a no ser que el usuario lo habilite intencionadamente.

Por otro lado, Elasticsearch no ha realizado cambios para el resto de opciones de seguridad como la comunicación segura mediante HTTPs, protocolos de encriptación o ficheros de auditoría. El resto de la configuración de seguridad corre por parte del usuario que deberá configurarlo de forma manual como ocurre en las versiones anteriores.

Por último, algo importante que añadir es que cuando se migra un configuración de 7.x a 8.x, Elasticsearch mantiene la configuración de seguridad que tuviera en ese momento el usuario, por tanto, las configuraciones automáticas de las que se ha hablado no servirían ya que solo se activan en la primera configuración del servicio.

9.- Conclusiones

Aunque este pentesting ha tenido como objetivo una única plataforma, probablemente existan muchas plataformas a las que le ocurra lo mismo, donde la configuración inicial no venga incluido una configuración de seguridad. Aunque realmente la plataforma deja a disposición del usuario un montón de herramientas de seguridad, estas no se usan ya que conllevan configuraciones adicionales y leer un poco la documentación, por lo que da a lugar a que el usuario que lo que busca es iniciar el servicio y usarlo, no se molesta en indagar qué tipo de configuraciones puedan existir, las cuales probablemente puedan serle útiles.

Por tanto, todo esto implica que la mayoría de los usuarios inicien servidores que están expuestos a internet los cuales son muy vulnerables, y todo esto, se hace de forma inconsciente, sin comprobar que realmente el servicio que se va a exponer es seguro.

En mi opinión, da mucho que pensar si realmente la gente tiene en cuenta el riesgo de exponer un servicio a internet, teniendo en cuenta que contiene información sensible. Durante la realización del pentesting, aunque descubrí la cantidad de cosas que se podían hacer al no estar habilitado nada sobre la seguridad, no veía factible realizar todo esto a nivel real, pero para mi sorpresa, la mayoría de los servicios que fuí encontrando estaban sin ningún tipo de configuración de seguridad. He encontrado servicios donde se guardan logs que provienen de otro sistemas, el servicio de archivos cloud de una empresa, base de datos de un hospital… y un largo etc. Siendo sincero, lo que más me sorprende de haber realizado este proyecto es que no se estén realizando robos de información a una velocidad desmesurada.

BIBLIOGRAFÍA

[1] Licencias en elasticsearch: https://www.elastic.co/es/subscriptions

[2] Seguridad en los endpoints y como configurarlo: https://www.elastic.co/guide/en/security/current/endpoint-security.html , https://www.elastic.co/guide/en/security/current/install-endpoint.html

[3] Alertas externas: https://www.elastic.co/guide/en/security/current/external-alerts.html

[4] Bloques de reglas: https://www.elastic.co/guide/en/security/current/building-block-rule.html

[5] Roles y autenticación: https://www.elastic.co/guide/en/elasticsearch/reference/7.14/built-in-roles.html

[6] Configurar seguridad básica en elasticsearch: https://www.elastic.co/guide/en/elasticsearch/reference/current/security-basic-setup.html

[7] Encriptación de las comunicaciones: https://www.elastic.co/guide/en/elasticsearch/reference/6.8/encrypting-communications.html

[8] Ficheros de configuración para la seguridad de elasticsearch: https://www.elastic.co/guide/en/elasticsearch/reference/current/security-files.html

[9] Configarición inicial de elasticsearch: https://www.elastic.co/guide/en/kibana/7.14/deb.html#deb-repo

[10] Definición de shodan en la Wikipedia: https://www.shodan.io/

[11] Que es nextcloud: https://nextcloud.com.es/

[12] Documentación de elasticsearch 7.x: https://www.elastic.co/guide/index.html?ultron=B-Stack-Trials-EMEA-S-Exact&gambit=Stack-Documentation&blade=adwords-s&hulk=paid&Device=c&thor=elasticsearch

%20documentation&gclid=Cj0KCQjw8p2MBhCiARIsADDUFVEbae60Z4p_oX3kBZUZz01acYHiy9RrhAT5pAo4LQCNa1RmLJ3EUkcaAp9uEALw_wcB

[13] Documentación de Elasticsearch para 8.x: https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html