Universitat Oberta de Catalunya

uoc.edu

Despliegue de plataforma de seguridad basada en Elastic Security

Alumno: Romano Piccio-Marchetti Prado
Plan de Estudios: MISTIC
Área: Seguridad Empresarial

Consultor: Jordi Guijarro Olivares
Profesor Responsable: Victor Garcia Font

Fecha: 01/06/2021

Esta obra está sujeta a una licencia Creative Commons CC BY-NC-ND 3.0 ES

FICHA DEL TRABAJO FINAL

Título del trabajo:	Despliegue de plataforma de seguridad basada en Elastic Security
Nombre del autor:	Romano Piccio-Marchetti Prado
Nombre del consultor:	Jordi Guijarro Olivares
Nombre del PRA:	Victor García Font
Fecha de entrega:	06/2021
Titulación:	MISTIC
Área del Trabajo Final:	Seguridad Empresarial
Idioma del trabajo:	Castellano
Palabras clave:	Endpoint Elastic SIEM

Resumen del Trabajo

Este Trabajo busca determinar si Elastic Security es un buen candidato para que una organización despliegue una solución de seguridad que le permita dotarse de forma sencilla de capacidades de detección, prevención y análisis de incidentes de seguridad.

Para ello se propone desplegar un entorno de evaluación donde poder probar en detalle las diferentes funcionalidades que ofrece la solución y que luego sea fácilmente trasladable a una infraestructura de tipo cloud que tuviera la organización.

Tras el despliegue del entorno de evaluación y la realización de las pruebas se puede concluir que Elastic Security es efectivamente una buena alternativa para que una organización implemente una solución de este tipo:

- Ofrece capacidades de detección y protección para endpoint que complementan las soluciones tradicionales de antivirus con la recolección de telemetría y la detección basada en comportamiento.
- Ofrece capacidades de detección de posibles incidentes de seguridad bastante completas, desde consultas simples a reglas complejas basadas en EQL o en detección de anomalías.
- Ofrece capacidades de análisis de los posibles incidentes de seguridad, con visualizaciones avanzadas y capacidades de integración con los sistemas de ticketing de la organización.
- Y además con Fleet se simplifica en gran medida la ingestión y normalización de fuentes de información externas, tanto de productos y soluciones de terceros como propietarios de la propia organización.

Abstract

This dissertation tries to determine if Elastic Security is a good candidate for an organization to deploy a security solution to enable in an easy way capabilities for detection, prevention, and analysis of security incidents.

This will be done through the deployment of a test environment where those capabilities can be tested. This environment must be easily transferable to a cloud-type infrastructure owned by the organization.

After deploying the test environment and performing extensive testing of the different features offered by the solution, the conclusion is that Elastic Security is indeed a good candidate:

- It has detection and protection features that complement traditional antivirus solutions incorporating centralized endpoint telemetry and behavior-based malware detection.
- It has extensive security incidents detection features, from simple queries over Elasticsearch indices to complex EQL expressions or anomaly detection jobs.
- It has analysis features, including advance event visualizations as well as integration features to enable case synchronization with some ticketing systems.
- And all of this with Fleet, a centralized agent management solution that simplifies deployment of endpoint protection as well as ingestion and normalization of data sources, both internal and third-party.

ÍNDICE

1. Introducción 1
1.1 Contexto y justificación del Trabajo 1
1.2 Objetivos del Trabajo 2
1.3 Enfoque y método seguido 3
1.4 Planificación del Trabajo 3
1.5 Breve sumario de productos obtenidos 5
1.6 Breve descripción de los otros capítulos de la memoria 5
2. Entorno de evaluación 6
3. Evaluación SIEM 13
3.1 Detección 13
3.1.1 Reglas Custom Query 18
3.1.2 Reglas Machine Learning 21
3.1.3 Reglas Threshold 22
3.1.4 Reglas Event Correlation 26
3.1.5 Reglas Indicator Match 30
3.2 Análisis 35
3.2.1 Líneas de Tiempo 35
3.2.2 Casos 37
3.3 Integración 39
3.3.1 Signals 39
3.3.2 Cases 40
4. Evaluación Endpoint Security 42
4.1 Fleet 42
4.2 Endpoint Security 44
5. Despliegue 50

6. Conclusiones 56

7. Glosario 59

8. Bibliografía 60

9. Anexos 65

9.1 Entorno de Demostración 65

LISTA DE FIGURAS

Figura 1. Planificación 3

Figura 2. Elastic Stack Home 11

Figura 3. Elastic Stack Management 12

Figura 4. Sobrescribir severidad. 14

Figura 5. Ajustes básicos de una regla. 15

Figura 6. Planificación de una regla. 16

Figura 7. Acciones de una regla. 17

Figura 8. Excepciones de una regla. 17

Figura 9. Alertas Custom Query 20

Figura 10. Mensaje de correo electrónico. 20

Figura 11. Alertas Machine Learning. 22

Figura 12. Alerta Threshold 25

Figura 13. Mensaje de correo electrónico. 26

Figura 14. Alerta de tipo Indicator Match. 29

Figura 15. Mensaje de correo electrónico. 30

Figura 16. Alerta de tipo Indicator Match. 34

Figura 17. Mensaje de correo electrónico. 34

Figura 18. Línea de Tiempo. 35

Figura 19. Consulta KQL en Línea de Tiempo. 36

Figura 20. Notas en Línea de Tiempo. 37

Figura 21. Casos. 38

Figura 22. Detalle de un Caso. 38

Figura 23. Configuración de Fleet. 42

Figura 24. Fleet Policies. 43

Figura 25. Fleet Agents. 44

Figura 26. Protección contra Malware. 45

Figura 27. Protección contra Ransomware. 45

Figura 28. Alertas por conexiones FTP. 48

Figura 29. Protección contra Malware. 49

Figura 30. Alertas de Endpoint Security. 49

Figura 31. Eventos con URLs de malware de abuse.ch. 65

Figura 32. Eventos de acceso del servidor proxy. 65

Figura 33. Regla para detectar accesos a URLs de malware. 66

Figura 34. Alerta de acceso a URL de malware. 66

1. Introducción

1.1 Contexto y justificación del Trabajo

Para que una organización pueda responder de forma rápida, efectiva y adecuada a los incidentes de seguridad de la información que se puedan producir en su ámbito de responsabilidad es fundamental que disponga de capacidades que la permitan detectar, prevenir y analizar posibles ataques, tanto a su infraestructura como a sus aplicaciones TI.

El despliegue de estas capacidades normalmente se hace desde dos enfoques complementarios. Por un lado, se despliegan capacidades tecnológicas específicas que permiten detectar y prevenir ciertos tipos de ataques. Por ejemplo, un producto antimalware para el correo electrónico o un firewall de aplicación para proteger al servidor web de la organización.

Y por otro, se despliegan capacidades de detección más generales basadas en el análisis de los eventos que pueden generar tanto esas tecnologías específicas como el resto de activos tecnológicos de la organización, como los sistemas operativos o las aplicaciones de negocio. Estos eventos se pueden analizar de forma conjunta y detectar así ataques más complejos.

Este tipo de capacidades de detección avanzada es a lo que habitualmente se suele denominar como SIEM. Combinan el almacenamiento de los eventos que se consideran relevantes desde el punto de vista de la seguridad, por lo general con fines de auditoria, con la detección de posibles ataques y el soporte de su investigación posterior por parte de analistas de seguridad.

Elastic Stack [1] lleva ya varios años siendo un referente en lo que a consolidación y visualización de logs se refiere. Beats [2] y Logstash [3] permiten extraer, transformar y cargar datos de un gran número de fuentes, Elasticsearch [4] permite almacenarlos y buscar de forma eficiente y Kibana [5] complementa la solución con visualizaciones avanzadas.

Por tanto, Elastic Security [6], solución construida sobre Elastic Stack que combina funcionalidades de EDR con funcionalidades de SIEM parece un buen candidato para que muchas organizaciones que ya estuvieran utilizando Elastic Stack se puedan dotar de forma relativamente sencilla de capacidades de detección, prevención y análisis de incidentes de seguridad.

El objetivo de este Trabajo es desplegar Elastic Security en un entorno de evaluación en donde poder analizar en detalle estas capacidades y que luego se pueda trasladar de forma sencilla a un entorno de tipo cloud donde sea posible escalarla según las necesidades de cada organización.

Si bien durante la preparación del plan de trabajo la versión de Elastic Security que había disponible era la 7.11, durante la primera fase de la ejecución se ha publicado la 7.12 [7], que será la que se evalúe finalmente. Se ha conseguido además una licencia educativa para poder evaluar funcionalidades que requieren de una subscripción de pago para su uso comercial.

1.2 Objetivos del Trabajo

Como ya se mencionaba en el apartado anterior, el objetivo de este Trabajo es desplegar Elastic Security en un entorno de evaluación en donde poder analizar en detalle las funcionalidades que ofrece la solución para detectar, prevenir y analizar posibles ataques tanto a la infraestructura como a las aplicaciones TI de una organización.

De cara a permitir un seguimiento adecuado del Trabajo, el plan docente de la asignatura contempla dos entregas parciales y una entrega final. Por tanto, parece también conveniente dividir el objetivo principal del Trabajo en objetivos secundarios más concretos que puedan estar asociados a las diferentes entregas del plan docente.

Además, si estos objetivos secundarios también están alineados con los aspectos clave mencionados en la propuesta original será también más sencillo determinar en qué medida se están teniendo en cuenta estos aspectos clave durante el desarrollo del Trabajo.

El primer objetivo secundario está relacionado con la necesidad de poder trasladar de forma sencilla el entono de evaluación a un entorno tipo cloud que sea posible escalar en función de las necesidades de cada organización. A priori parece que Elastic Cloud on Kubernetes [8] sobre Minikube [9] es una buena alternativa, pero es algo a verificar durante el desarrollo del Trabajo.

El segundo objetivo secundario está relacionado con el análisis de las capacidades de SIEM que ofrece Elastic Security. Para ello, se integrarán varias fuentes de información en el entorno de evaluación y se probarán las capacidades de detección y de análisis de la solución.

Se intentarán verificar tanto las capacidades de detección básicas basadas en consultas a Elasticsearch como las basadas en EQL, más enfocado a la seguridad y que permite establecer relaciones complejas entre eventos. Se analizará también el uso de feeds de terceros.

El tercer objetivo secundario está relacionado con las capacidades de integración de la solución de Elastic Security con otros sistemas de pueda tener la organización, como por ejemplo, envío de correos electrónicos, mensajería instantánea, sistemas de tickets o sistemas de orquestación SOAR.

Estas acciones externas no están incluidas en la subscripción gratuita de Elastic Security, por lo que aunque se disponga de una licencia de evaluación en donde si están disponibles, se intentará también evaluar si al menos es posible hacer algo parecido empleando otras funcionalidades si disponibles, como por ejemplo mediante consultas periódicas al API REST.

Por último, el cuarto objetivo secundario está relacionado con el análisis de las capacidades de endpoint que ofrece Elastic Security. Para ello se instalará el agente de Elastic Endpoint Security [10] en al menos un equipo con Microsoft Windows y otro equipo con Linux y se verificarán las funcionalidades principales del producto.

1.3 Enfoque y método seguido

El tema del Trabajo no deja mucho espacio a valorar estrategias puesto que se parte de un producto ya seleccionado como es Elastic Security y se establece en la propuesta original que el entorno de evaluación debe ser trasladable de forma sencilla a entornos cloud.

Lo que si se ha analizado han sido las alternativas para desplegar ese entorno de evaluación, si se optaba por instalación local o por instalación en cloud pública y si se usaba una solución de orquestación de contenedores como puede ser Kubernetes o una instalación más *ad-hoc*.

Tanto por comodidad como por economía de medios, al final se ha optado por una instalación local en un equipo doméstico. Y en cuanto al tipo de instalación, se ha optado por utilizar Kubernetes [12], puesto que está soportado por Elastic mediante su operador Elastic Cloud on Kubernetes y es un enfoque que puede facilitar mucho la migración a otros entornos.

No se ha considerado el uso de Elastic Cloud [13] ya que como solución de referencia parece más limitante y además, tiene un coste asociado. Y por motivos parecidos, tampoco se ha contemplado el uso de OpenShift [14]. No obstante, si luego alguna organización prefiere utilizar alguna de estas soluciones tampoco debería suponer un problema.

En cuanto a cómo enfocar la evaluación, en la mayoría de las ocasiones se ha optado por implementar casos de uso sencillos que permitan ilustrar y entender bien las capacidades de Elastic Security frente a escenarios más realistas con productos comerciales involucrados, ya que introducen complejidad añadida sin aportar otras ventajas adicionales.

1.4 Planificación del Trabajo

A continuación se presentan las tareas e hitos más relevantes de este Trabajo en forma de diagrama de Gantt, en donde se puede ver tanto su duración estimada como sus dependencias. El diagrama se ha realizado con la herramienta GanttProject [11], con licencia GPL.

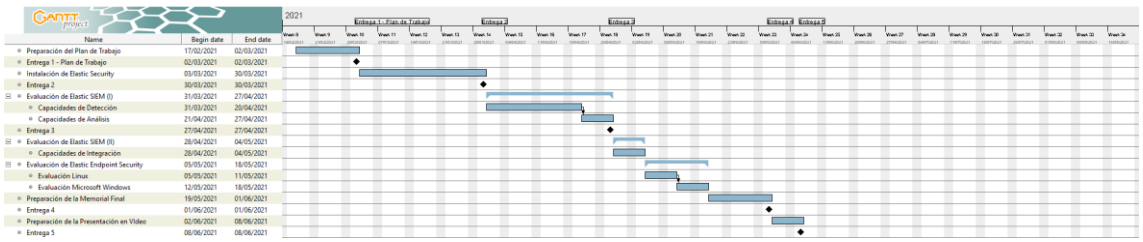

Figura 1. Planificación

Como se puede observar en la figura, se han intentado alinear las tareas con los objetivos del Trabajo ya expuestos anteriormente así como asociarlas a los diferentes hitos identificados, que además coinciden con las entregas definidas en el plan docente de la asignatura.

Se han realizado algunos ajustes respecto a la planificación original presentada en el Plan de Trabajo. Se ha adelantado la evaluación de las capacidades que Elastic Security ofrece como SIEM y así tener la posibilidad si fuera necesario de deslizar alguna actividad para la siguiente entrega.

En cuanto a los recursos necesarios para poder completar estas tareas, lo principal es disponer de un ordenador personal con capacidad suficiente para preparar, ejecutar y documentar el Trabajo. Para preparar y documentar el Trabajo se ha necesitado al menos el siguiente software:

Producto	**Versión**
Microsoft Windows	10
Google Chrome	88
Microsoft Word	2008
GanttProject	2.8.10

Para ejecutar el Trabajo se ha necesitado el siguiente software:

Producto	**Versión**
Microsoft Windows	10
Oracle VirtualBox	6.1.18
Debian Linux	10
Docker Engine	20.10.5
kubectl	1.20.5
Minikube	1.18.1
certbot	0.31
Elastic Cloud on Kubernetes	1.5.0
Elasticsearch	7.12.0
Kibana	7.12.0
Elastic Security	7.12.0

El coste de un ordenador personal que permita completar las tareas del Trabajo puede estar en torno a los 1.000 euros. Debe ser un equipo que tenga al menos una CPU Intel Core i7 de cuatros núcleos o equivalente y al menos 16 GB de memoria RAM.

Además del ordenador personal, será también necesaria una conexión a Internet para las tareas de investigación, para la descarga de software y para algunos casos de uso como la evaluación de *feeds* de amenazas. El coste asociado a esta conexión se estima en unos 50 euros al mes.

1.5 Breve sumario de productos obtenidos

El producto de este Trabajo es esta memoria, que se puede ver desde varios puntos de vista. Por un lado, se puede ver como una guía práctica sobre cómo desplegar de forma rápida un entorno de Elastic Stack sobre Kubernetes que permita probar la solución.

Por otro lado, se puede ver como un análisis detallado de las capacidades que ofrece Elastic Security, tanto como solución EDR para detección y protección de *endpoints* como solución SIEM para detección y análisis de posibles incidentes de seguridad.

Por último, se puede ver también como una serie de recomendaciones que una organización debería tener en cuenta antes de desplegar Elastic Security, tanto genéricas relacionadas con Elastic Stack como más específicas sobre las capacidades propias de Elastic Security.

1.6 Breve descripción de los otros capítulos de la memoria

La memoria de este Trabajo se ha estructurado en ocho capítulos, siendo esta Introducción el primero de ellos. En el segundo capítulo se describe en detalle el entorno de evaluación que se ha empleado para realizar este Trabajo, con instrucciones detalladas que permitan desplegar un entorno similar.

El tercer capítulo presenta la evaluación de las capacidades de SIEM de Elastic Security en sus tres ejes principales: la detección de posibles incidentes de seguridad, el análisis de estos posibles incidentes y la integración de la solución con otros sistemas y plataformas.

El cuatro capítulo presenta la evaluación de las capacidades de seguridad para endpoint de Elastic Security, tanto de gestión del parque de equipos con Fleet como de obtención de telemetría y de protección activa frente a determinadas amenazas, como malware o ransomware, con Endpoint Security.

En el quinto capítulo se presentan las consideraciones principales que una organización debería tener en cuenta para desplegar Elastic Security, incluyendo tanto conocimientos previos recomendados de Elastic Stack como aspectos concretos de la solución de Elastic Security que conviene tener presentes.

Por último, en el sexto se presentan las conclusiones del Trabajo, el séptimo es un glosario de términos utilizados en la memoria y el octavo es la bibliografía, en donde se recogen las principales fuentes consultadas durante la realización del Trabajo y referencias donde se puede ampliar información.

2. Entorno de evaluación

El objetivo de este Trabajo es desplegar Elastic Security en un entorno de evaluación en donde poder analizar en detalle las diferentes funcionalidades que ofrece la solución, pero que debe ser trasladable de forma sencilla a un entorno de tipo cloud que sea posible escalar en función de las necesidades concretas de cada organización.

Tanto por comodidad como por economía de medios, la propuesta para este entorno de evaluación consiste en utilizar como anfitrión un equipo doméstico de propósito general con una CPU Intel Core i7-6700 a 3.40 GHz, el cual tiene 4 núcleos y 8 procesadores, 16 GB de memoria RAM y sistema operativo Microsoft Windows 10 Enterprise.

Para no interferir con otros usos del equipo anfitrión y como éste tiene instalado el software de virtualización Oracle VirtualBox [15] en su versión 6.1.18, se creará además una máquina virtual con sistema operativo Linux Debian 10.8.0 y las Guest Additions de VirtualBox, asignándola 4 procesadores de los 8 de los que dispone el equipo anfitrión y 8 GB de los 16 GB disponibles de RAM.

Docker Engine

En esta máquina virtual se instalará también la versión 20.10.5 del software de contenerización Docker Engine [16], que es el que usará Minikube como driver de virtualización. Para ello se deben seguir las instrucciones oficiales para sistemas operativos Linux Debian empleando el gestor de paquetes APT y el repositorio oficial de Docker.

<u>Añadir repositorio APT</u>

Actualizar el índice de paquetes de APT

```
$ sudo apt-get update
```

Instalar los paquetes necesarios para usar repositorios APT sobre HTTPS

```
$ sudo apt-get install \
    apt-transport-https \
    ca-certificates \
    curl \
    gnupg \
    lsb-release
```

Añadir la clave GPG oficial de Docker

```
$ curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor \
    -o /usr/share/keyrings/docker-archive-keyring.gpg
```

Añadir el repositorio APT estable de Docker

```
$ echo "deb [arch=amd64 \
    signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] \
    https://download.docker.com/linux/debian \
    $(lsb_release -cs) stable" | sudo tee \
    /etc/apt/sources.list.d/docker.list > /dev/null
```

Instalar Docker Engine

Actualizar el índice de paquetes de APT

```
$ sudo apt-get update
```

Instalar la última versión de los paquetes de Docker Engine y de containerd

```
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
```

Verificar que Docker Engine está instalado correctamente

```
$ sudo docker run hello-world
```

Es posible ejecutar contenedores Docker sin necesidad de ser `root`, solo hace falta tener acceso al socket UNIX que el producto crea en `/var/run/docker.sock` y que suele significar pertenecer al grupo `docker`, lo cual se puede hacer con el siguiente comando:

```
$ sudo usermod -a -G docker romano
```

Si el usuario que quiere ejecutar contenedores Docker es `romano`. Antes de que sea realmente efectivo es necesario reevaluar los grupos a los que pertenece el usuario, por ejemplo saliendo de la shell y volviendo a acceder al sistema.

kubectl

También se debe instalar la versión 1.20.5 de kubectl [17], la herramienta de línea de comandos de Kubernetes. Para ello se deben seguir las instrucciones oficiales para sistemas operativos Linux Debian empleando el gestor de paquetes APT y los repositorios oficiales de Kubernetes.

Añadir repositorio APT

Actualizar el índice de paquetes de APT

```
$ sudo apt-get update
```

Instalar los paquetes necesarios para usar repositorios APT sobre HTTPS

```
$ sudo apt-get install -y apt-transport-https ca-certificates curl
```

Añadir la clave GPG oficial de Google Cloud:

```
$ sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg \
    https://packages.cloud.google.com/apt/doc/apt-key.gpg
```

Añadir el repositorio APT de Kubernetes:

```
$ echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] \
    https://apt.kubernetes.io/ kubernetes-xenial main " | sudo tee \
    /etc/apt/sources.list.d/kubernetes.list
```

Instalar kubectl

Actualizar el índice de paquetes de APT

```
$ sudo apt-get update
```

Instalar la última versión de kubectl

```
$ sudo apt-get install -y kubectl
```

Activar autocompletado de kubectl

```
$ kubectl completion bash | sudo tee /etc/bash_completion.d/kubectl
```

minikube

Para disponer de una instalación local de Kubernetes se instalará también en esta máquina virtual la versión 1.18.1 de minikube. Para ello se deben seguir las instrucciones oficiales para sistemas operativos Linux Debian con arquitectura amd64:

Descargar minikube

```
$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube_latest_amd64.deb
```

Instalar minikube

```
$ sudo dpkg -i minikube_latest_amd64.deb
```

Activar autocompletado de minikube

```
$ minikube completion bash | sudo tee /etc/bash_completion.d/minikube
```

Tras esto ya debería ser posible arrancar un clúster de Minikube. Por ejemplo, para arrancar un clúster de tres nodos asociado al perfil `tfm`:

```
$ minikube start --nodes=3 --profile='tfm'
```

Si en algún momento es necesario consultar los clústeres y su estado, se puede hacer con el siguiente comando:

```
$ minikube profile list

|---------|-----------|---------|--------------|------|---------|---------|-------|
| Profile | VM Driver | Runtime |      IP      | Port | Version | Status  | Nodes |
|---------|-----------|---------|--------------|------|---------|---------|-------|
| tfm     | docker    | docker  | 192.168.49.2 | 8443 | v1.20.2 | Running |     3 |
|---------|-----------|---------|--------------|------|---------|---------|-------|
```

Una vez esté el clúster de Kubernetes operativo es una buena práctica crear un *namespace* específico para todos los recursos asociados:

```
$ kubectl create namespace tfm
```

Y configurarlo como *namespace* por defecto:

```
$ kubectl config set-context --current --namespace=tfm
```

Elastic Cloud on Kubernetes

La instalación de la versión 1.5.0 de ECK consiste en aplicar sobre el clúster de Kubernetes el operador que proporciona Elastic, con sus reglas RBAC y con sus definiciones de recursos personalizadas.

```
$ kubectl apply -f https://download.elastic.co/downloads/eck/1.5.0/all-in-one.yaml
```

Para evitar problemas posteriores de conexión, antes de instalar Elasticsearch y Kibana primero se deben generar un certificado y una clave para proteger las comunicaciones TLS con el exterior. Para ello se va a utilizar la autoridad de certificación gratuita Let's Encrypt [18] a través de la herramienta certbot [19] de la EFF [20].

Lo primero es instalar certbot:

```
$ sudo apt-get install certbot
```

Una vez que la utilidad está instalada, ya se puede solicitar el certificado. Se va a solicitar un certificado *wildcard* para el dominio tfm.kanoon.eu, acreditando la propiedad del dominio añadiendo una entrada DNS de tipo TXT.

```
$ sudo certbot certonly --manual --preferred-challenges dns -d "*.tfm.kanoon.eu"
```

Para poder utilizar estos certificados en Elasticsearch y en Kibana es necesario primero crear un secreto en el clúster de Kubernetes:

```
$ kubectl create secret generic tfm.kanoon.eu \
    --from-file=ca.crt=ca.crt --from-file=tls.crt=tls.crt --from-file=tls.key=tls.key
```

El fichero ca.crt debe contener la CA Root de Let's Encrypt:


```
-----BEGIN CERTIFICATE-----
MIIDSjCCAjKgAwIBAgIQRK+wgNajJ7qJMDmGLvhAazANBgkqhkiG9w0BAQUFADA/
MSQwIgYDVQQKExtEaWdpdGFsIFNpZ25hdHVyZSBUcnVzdCBDby4xFzAVBgNVBAMT
DkRTVCBSb290IENBIFgzMB4XDTAwMDkzMDIxMTIxOVoXDTIxMDkzMDE0MDExNVow
PzEkMCIGA1UEChMbRGlnaXRhbCBTaWduYXR1cmUgVHJ1c3QgQ28uMRcwFQYDVQQD
Ew5EU1QgUm9vdCBDQSBYMzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
AN+v6ZdQCINXtMxiZfaQguzH0yxrMMpb7NnDfcdAwRgUi+DoM3ZJKuM/IUmTrE4O
rz5Iy2Xu/NMhD2XSKtkyj4zl93ewEnu1lcCJo6m67XMuegwGMoOifooUMM0RoOEq
OLl5CjH9UL2AZd+3UWODyOKIYepLYYHsUmu5ouJLGiifSKOeDNoJjj4XLh7dIN9b
xiqKqy69cK3FCxolkHRyxXtqqzTWMIn/5WgTe1QLyNau7Fqckh49ZLOMxt+/yUFw
7BZy1SbsOFU5Q9D8/RhcQPGX69Wam40dutolucbY38EVAjqr2m7xPi71XAicPNaD
aeQQmxkqtilX4+U9m5/wAl0CAwEAAaNCMEAwDwYDVR0TAQH/BAUwAwEB/zAOBgNV
HQ8BAf8EBAMCAQYwHQYDVR0OBBYEFMSnsaR7LHH62+FLkHX/xBVghYkQMA0GCSqG
SIb3DQEBBQUAA4IBAQCjGiybFwBcqR7uKGY3Or+Dxz9LwwmglSBd49lZRNI+DT69
ikugdB/OEIKcdBodfpga3csTS7MgROSR6cz8faXbauX+5v3gTt23ADq1cEmv8uXr
AvHRAosZy5Q6XkjEGB5YGV8eAlrwDPGxrancWYaLbumR9YbK+rlmM6pZW87ipxZz
R8srzJmwN0jP41ZL9c8PDHIyh8bwRLtTcm1D9SZImlJnt1ir/md2cXjbDaJWFBM5
JDGFoqgCWjBH4d1QB7wCCZAA62RjYJsWvIjJEubSfZGL+T0yjWW06XyxV3bqxbYo
Ob8VZRzI9neWagqNdwvYkQsEjgfbKbYK7p2CNTUQ
-----END CERTIFICATE-----
```

Los otros dos ficheros se pueden obtener del directorio donde la utilidad certbot almacena los certificados y claves:

```
tls.key -> /etc/letsencrypt/live/tfm.kanoon.eu/privkey.pem
tls.crt -> /etc/letsencrypt/live/tfm.kanoon.eu/fullchain.pem
```

Elasticsearch

Teniendo ECK ya desplegado en un clúster de Kubernetes, crear un clúster de Elasticsearch 7.12.0 de tres nodos es sencillo. Este clúster utilizará el certificado y la clave que habíamos creado previamente.

```
$ cat <<EOF | kubectl apply -f -
apiVersion: elasticsearch.k8s.elastic.co/v1
kind: Elasticsearch
metadata:
  name: tfm
spec:
  version: 7.12.0
  nodeSets:
  - name: default
    count: 3
    config:
      node.store.allow_mmap: false
  http:
    tls:
      certificate:
        secretName: tfm.kanoon.eu
EOF
```

Podemos ver el estado de los pods con el siguiente comando:

```
$ kubectl get pods

NAME                 READY   STATUS    RESTARTS   AGE
tfm-es-default-0     1/1     Running   0          2m25s
tfm-es-default-1     1/1     Running   0          2m25s
tfm-es-default-2     1/1     Running   0          2m25s
```

Kibana

Al igual que con Elasticsearch, teniendo ECK ya desplegado en el clúster de Kubernetes, crear un clúster de Kibana de tres nodos sobre el clúster de Elasticsearch desplegado anteriormente es también sencillo. Este clúster utilizará también el certificado y la clave que habíamos creado previamente.

```
$ cat <<EOF | kubectl apply -f -
apiVersion: kibana.k8s.elastic.co/v1
kind: Kibana
metadata:
  name: tfm
spec:
  version: 7.12.0
  count: 3
  elasticsearchRef:
    name: tfm
  http:
    tls:
      certificate:
        secretName: tfm.kanoon.eu
EOF
```

Podemos ver el estado de los pods con el siguiente comando:

```
$ kubectl get pods

NAME                      READY   STATUS    RESTARTS   AGE
tfm-kb-7778b5fcd-97xlw    1/1     Running   0          3m30s
tfm-kb-7778b5fcd-d2qct    1/1     Running   0          3m30s
tfm-kb-7778b5fcd-fhrn5    1/1     Running   0          3m30s
```

Para acceder a Kibana lo primero es obtener la contraseña del usuario elastic:

```
$ kubectl get secret tfm-es-elastic-user -o=jsonpath='{.data.elastic}' | base64 --decode; echo
```

Hay que redirigir también el puerto HTTPS de Elasticsearch y de Kibana para que sean accesibles desde el anfitrión y desde otras máquinas virtuales:

```
$ kubectl port-forward --address 192.168.98.200 service/tfm-es-http 9200:9200
$ kubectl port-forward --address 192.168.98.200 service/tfm-kb-http 5601:5601
```

Con eso ya se debería poder acceder a Kibana:

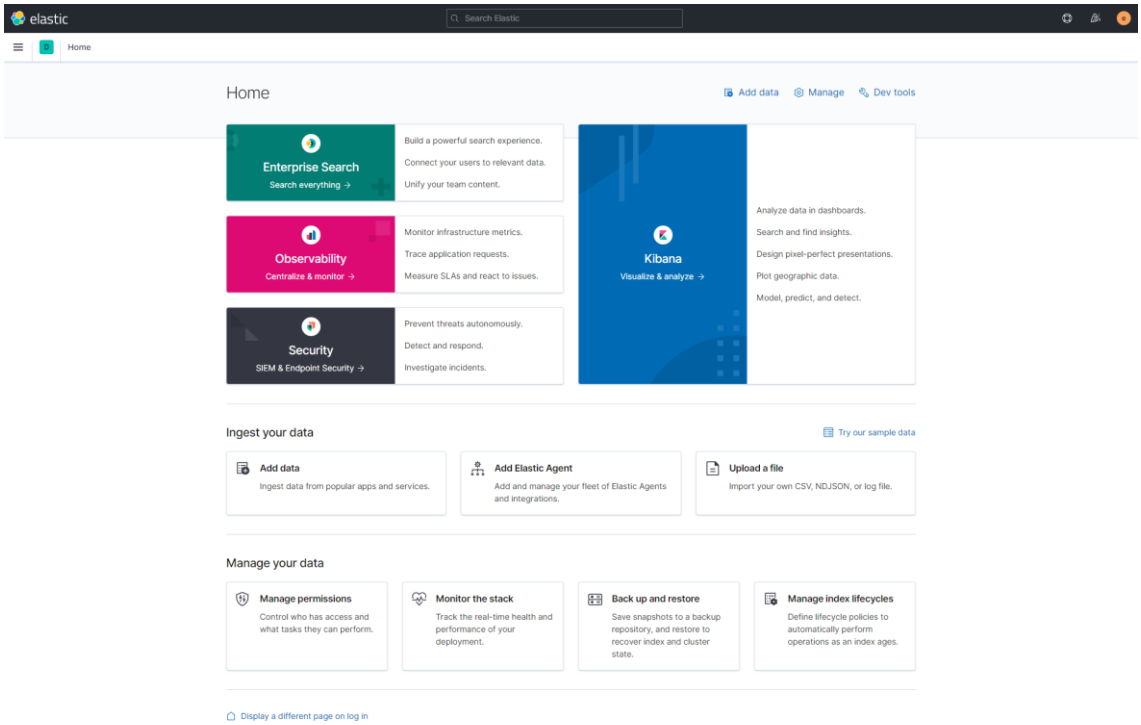

Figura 2. Elastic Stack Home

Al pulsar sobre la opción Manage se accederá a Stack Management, en donde entre otras cosas, podremos comprobar que efectivamente hemos instalado la versión 7.12.0 de Elastic Stack:

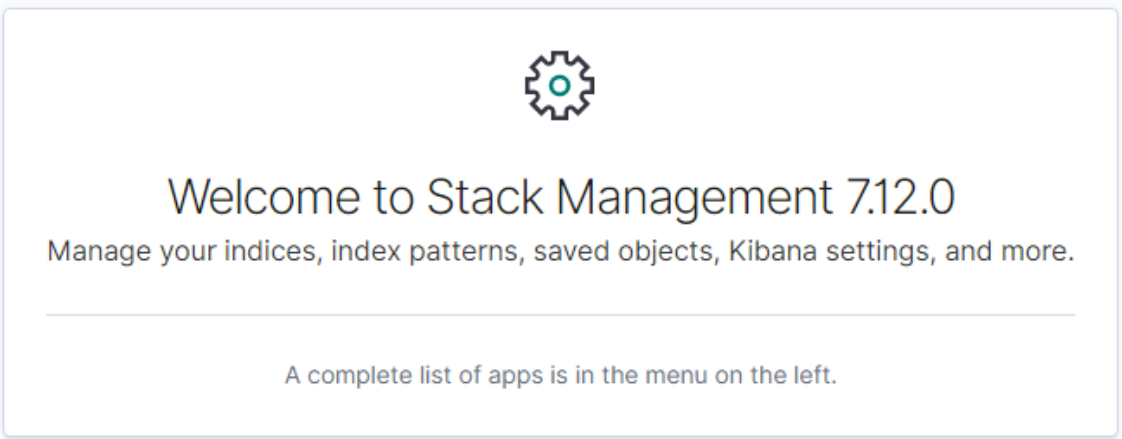

Figura 3. Elastic Stack Management

El siguiente paso es añadir la licencia de evaluación que permite utilizar las capacidades del producto que habitualmente tienen coste asociado. Esto se puede hacer desde la consola de desarrollo haciendo una llamada a la API:

```
PUT /_license
{
  "license": {
    "uid": "407fd959-021f-42e0-9234-18460d617e48",
    "type": "trial",
    "issue_date_in_millis": 1614816000000,
    "expiry_date_in_millis": 1630799999999,
    "max_nodes": 100,
    "issued_to": "Nonprofit / Educational (non-production environments)",
    "issuer": "API",
    "signature": "**********",
    "start_date_in_millis": 1614816000000
  }
}
```

3. Evaluación SIEM

3.1 Detección

Las capacidades de detección de Elastic Security se basan en dos conceptos principales: reglas de detección y alertas de detección. Una regla de detección es una regla que se ejecuta periódicamente buscando eventos que cumplan con ciertos criterios. Cuando se encuentran eventos que cumplen con esos criterios, entonces se crea una alerta de detección.

Elastic Security incluye un conjunto de reglas de detección predefinidas que por defecto no están cargadas pero que se pueden cargar de forma sencilla en cualquier momento desde la interfaz de usuario. Si se opta por cargarlas, hay que tener en cuenta que se cargan todas desactivadas salvo la regla que trata alertas de Endpoint Security.

El conjunto de reglas incluidas en cada versión de Elastic Security es una instantánea del repositorio público [21] en donde se mantienen las reglas de detección. En dicho repositorio, al que cualquiera que cumpla con los criterios establecidos puede contribuir, se pueden encontrar nuevas reglas y modificaciones de reglas existentes que todavía no han sido publicadas.

Para entender mejor los diferentes tipos de reglas de detección que se pueden definir en Elastic Security, en esta evaluación se crearán reglas específicas en lugar de usar el conjunto de reglas predefinidas. En la versión 7.12 se pueden crear reglas de cinco tipos diferentes:

- **Custom Query**: Se crea una alerta de detección cuando se encuentran documentos que cumplen con los criterios establecidos. La consulta se puede especificar como una expresión KQL [22] o Lucene [23].

- **Machine Learning**: Se crea una alerta de detección cuando un trabajo *machine learning* descubre una anomalía por encima de un umbral configurado. Solamente disponible en licencias de tipo *platinum*.

- **Threshold**: Se crea una alerta de detección cuando se encuentran más de ciertos documentos con el mismo valor en un determinado campo. Se crearán tantas alertas como valores por encima del umbral.

- **Event Correlation**: Se crea una alerta de detección cuando se encuentran documentos que cumplen con los criterios establecidos. La consulta se especifica como una expresión EQL [24].

- **Indicator Match**: Se crea una alerta de detección cuando se encuentran documentos que tienen campos con ciertos valores que se obtienen de una segunda consulta. Se suelen usar para integrar *feeds* de terceros.

Aunque cada tipo de regla requiere un configuración distinta hay secciones que son comunes a todos los tipos de reglas. En la sección About se pueden configuran ciertos ajustes de la regla, tanto básicos como avanzados.

Los básicos son los siguientes:

a) **Nombre**: El nombre de la regla.

b) **Descripción**: Una descripción de lo que hace la regla.

c) **Severidad por defecto**: La severidad por defecto de la regla.

- Low: Alertas que por lo general no son incidentes de seguridad.
- Medium: Alertas que requieren investigación.
- High: Alertas que requieren investigación inmediata.
- Critical: Alertas que por lo general son incidentes de seguridad.

d) **Sobrescribir severidad**: Opcional. Se pueden usar los valores de ciertos campos de los eventos para determinar la severidad de la alerta.

Figura 4. Sobrescribir severidad.

Si hay más de una coincidencia se usará la severidad más grave y si no hay coincidencias, se usara la severidad por defecto.

e) **Puntuación de riesgo por defecto**: Un valor numérico entre 0 y 100 y que debe estar relacionado con la severidad. En general:

- De 0 a 21 para alertas de severidad baja (Low).
- De 22 a 47 para alertas de severidad media (Medium).
- De 48 a 73 para alertas de severidad alta (High).
- De 74 a 100 para alertas de severidad crítica (Critical).

f) **Sobrescribir puntuación de riesgo**: Opcional. Se puede usar el valor de un campo de los eventos para sobrescribir la puntuación de riesgo de la alerta.

g) **Etiquetas**: Opcional. Permiten categorizar las reglas, facilitando así tanto su búsqueda como su filtrado.

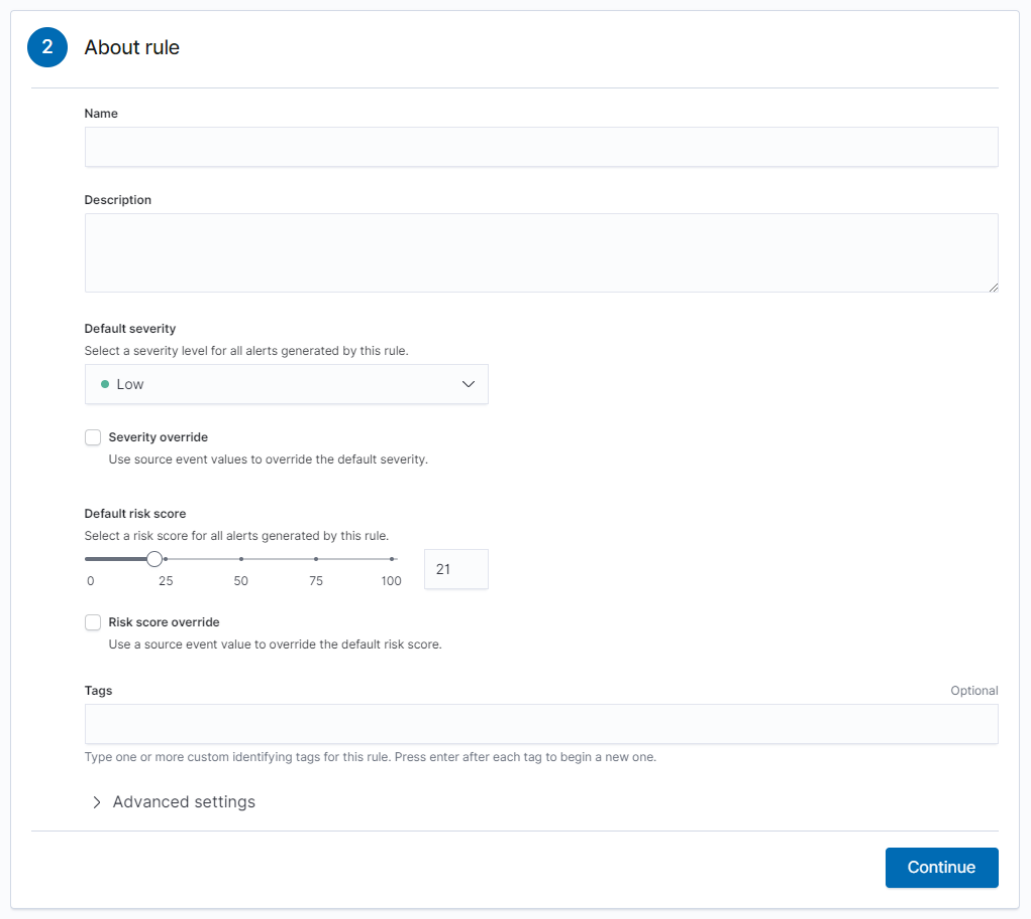

Figura 5. Ajustes básicos de una regla.

Los ajustes avanzados son los siguientes:

a) **Referencias**: Opcional. Referencias en forma de URL a información que pueda ser relevante para la regla.

b) **Ejemplos de falsos positivos**: Opcional. Escenarios habituales en los que se pueden producir falsos positivos.

c) **Amenazas MITRE ATT&CK**: Opcional. Tácticas, técnicas y subtécnicas de la base de conocimiento ATT&CK [25] de MITRE [26].

d) **Guía de investigación**: Opcional. Información para que los analistas investiguen las alertas generadas por la regla.

e) **Autor**: Opcional. Los autores de la regla.

f) **Licencia**: Opcional. La licencia de la regla.

g) **Excepciones de Endpoint Security**: Opcional. Permite añadir todas las excepciones de la regla de Endpoint Security a esta regla.

h) **Building block**: Opcional. Permite marcar la regla como de tipo building block, lo que hace que no aparezca en la interfaz de usuario. Se puede

usar para evitar ruido innecesario por alertas de severidad baja o para crear reglas de detección sobre alertas generadas por otras reglas.

i) **Sobrescribir nombre de regla**: Opcional. Se puede usar un campo del evento para que aparezca como nombre de la regla en la interfaz de usuario. Esto permite usar nombres más explicativos.

j) **Sobrescribir timestamp**: Opcional. Se puede usar un campo del evento como timestamp en lugar del campo por defecto @timestamp. Esto puede ser útil en ciertos escenarios para evitar perder alertas por retrasos en la ingesta de los eventos.

En la sección Schedule se puede configurar cada cuanto tiempo se quiere ejecutar la regla. Adicionalmente, se puede configurar un parámetro para extender la ventana de búsqueda de la regla.

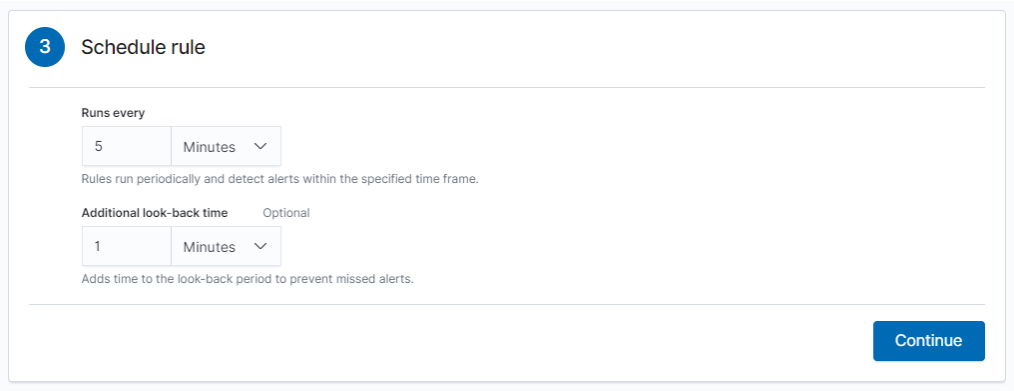

Figura 6. Planificación de una regla.

Si por ejemplo la regla está configurada para ejecutarse cada 5 minutos y este parámetro tiene un valor de 5 minutos, cada vez que se ejecute se analizarán los eventos de los últimos diez minutos.

Elastic recomienda que como mínimo ese parámetro se configure a 1 minuto y así evitar perder detecciones por pequeños retrasos en la ejecución de las reglas. Elastic Security evita la creación de alertas duplicadas debido al tiempo adicional de búsqueda hacia atrás.

Por último, en la sección Actions se puede configurar qué acciones de Kibana se quieren ejecutar cuando se genera una alerta. Estas acciones solo están disponibles en licencias *gold* o superiores, dependiendo del tipo de acción.

Lo primero será configurar la frecuencia con la que se quieren ejecutar las acciones: cada vez que se ejecuta la regla, cada hora, cada día o cada semana. Las acciones solo se ejecutarán si se han generado nuevas alertas, con independencia de la frecuencia que se configure.

Además, hay que seleccionar las acciones que se quieren ejecutar. Por cada acción se debe seleccionar el tipo, el conector concreto que se quiere emplear (puede existir previamente o se puede crear en este momento) así como los

campos específicos de cada tipo. En estos campos se pueden utilizar *placeholders* para incluir información tanto de la regla como de la alerta.

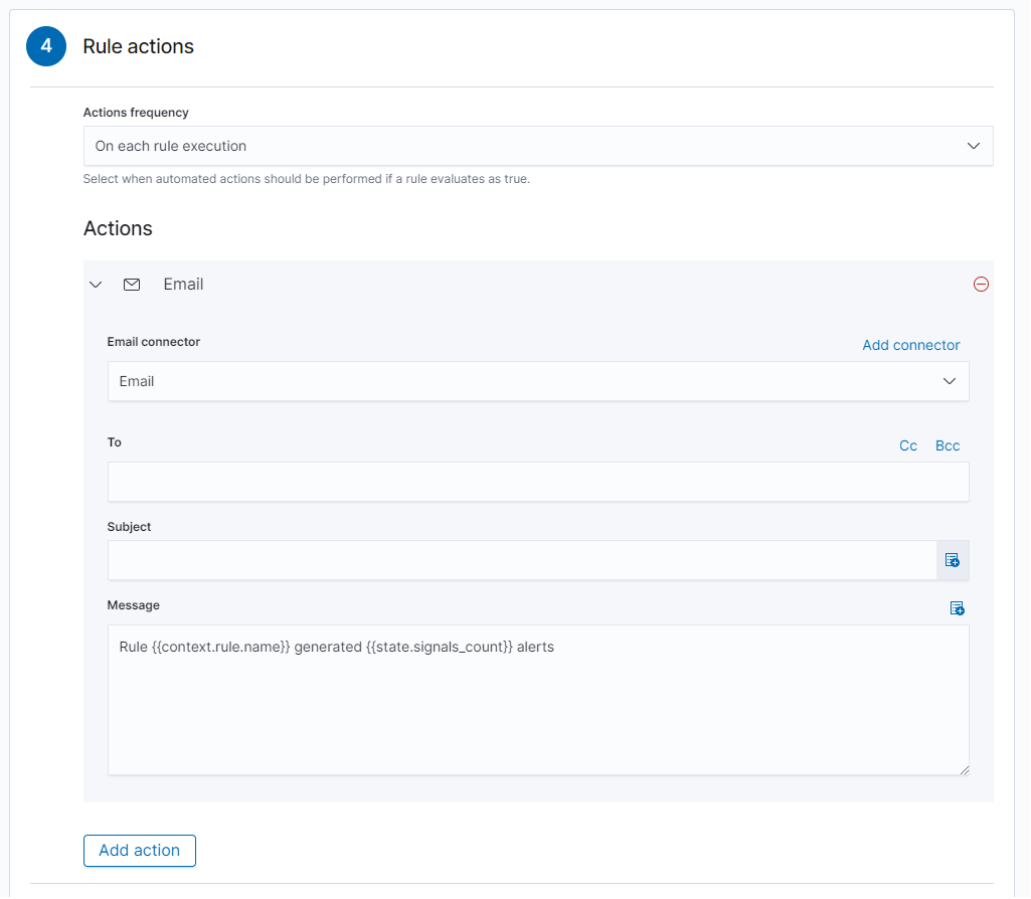

Figura 7. Acciones de una regla.

Además las reglas permiten definir excepciones. Una excepción es una combinación lógica de condiciones sobre los campos de los eventos que pueden generar una alerta que si se cumple, evita que se genere. Son útiles para refinar las reglas y reducir progresivamente el número de falsos positivos que puedan producir.

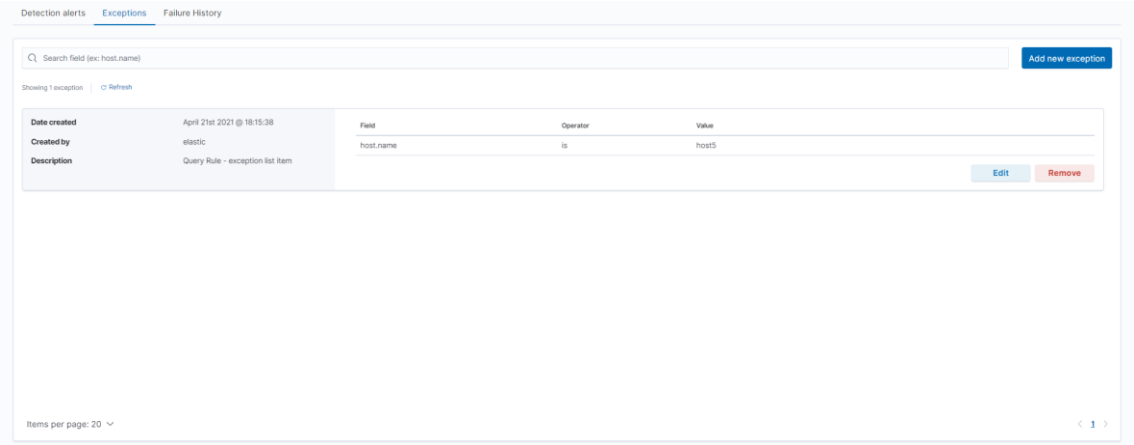

Figura 8. Excepciones de una regla.

3.1.1 Reglas Custom Query

Se crea una alerta de detección cuando se encuentran documentos que cumplen con los criterios establecidos, que pueden ser tanto expresiones KQL como expresiones Lucene. Para crear una regla de tipo Custom Query es necesario completar los siguientes campos:

a) **Index patterns**: Los patrones de los índices sobre los que se buscarán los documentos.

b) **Custom query**: Los filtros y la consulta que se utilizarán para obtener los documentos.

c) **Timeline template**: La plantilla de línea de tiempo a utilizar de la lista de plantillas de líneas de tiempo disponibles.

Para ilustrar este tipo de reglas de detección, primero se creará en el entorno de evaluación un index template `tfm-events` que aplicará sobre índices de tipo datastream que se llamen `tfm-events`. Este index template define los campos que pueden tener los documentos.

```
PUT /_index_template/tfm-events
{
    "index_patterns":[
        "tfm-events"
    ],
    "data_stream":{
    },
    "template":{
        "mappings":{
            "date_detection":false,
            "dynamic":"strict",
            "dynamic_templates":[
                {
                    "strings_as_keywords":{
                        "match_mapping_type":"string",
                        "mapping":{
                            "type":"keyword"
                        }
                    }
                }
            ],
            "properties":{
                "@timestamp":{
                    "type":"date",
                    "format":"yyyy-MM-dd'T'HH:mm:ss.SSSXXX"
                },
                "destination":{
                    "type":"object",
                    "properties":{
                        "ip":{
                            "type":"ip"
                        }
                    }
                },
                "host":{
                    "type":"object",
                    "properties":{
                        "name":{
                            "type":"keyword"
                        }
                    }
                },
                "source":{
                    "type":"object",
```

```
                    "properties":{
                        "ip":{
                            "type":"ip"
                        }
                    }
                }
            }
        }
    }
}
```

Se creará además una regla de detección con nombre Query Rule, con severidad Medium, con puntuación de riesgo 25 y con la siguiente definición:

a) **Index patterns**: `tfm-events`

b) **Filters**: Ninguno.

c) **Custom query**: `destination.ip: 193.24.23.13`

Esta regla de detección se ejecutará cada minuto con una ventana de 12 horas y tendrá además asociada una acción de tipo Email que enviará un correo cada vez que se creen alertas de detección con los siguientes parámetros:

a) **To**: `rpiccio@uoc.edu`

b) **Subject**: `New {{context.rule.name}} alerts`

c) **Message**: `Rule generated {{state.signals_count}} alerts`

A continuación se irán creando documentos en el índice `tfm-events` y se comprobará si tras la ejecución de la regla de detección se han generado nuevas alertas y se ha enviado el mensaje de correo electrónico esperado.

```
POST /tfm-events/_doc/
{
  "@timestamp": "2021-04-21T16:00:00.000Z",
  "destination":{
    "ip":"193.24.23.12"
  },
  "host":{
    "name":"host1"
  },
  "source":{
    "ip":"217.12.12.1"
  }
}
```

Tras insertar este evento y esperar a que se ejecute la regla de detección se comprueba que no se han creado alertas de detección, ya que la query configurada no produce resultados.

```
POST /tfm-events/_doc/
{
  "@timestamp": "2021-04-21T16:00:00.000Z",
  "destination":{
    "ip":"193.24.23.13"
  },
  "host":{
    "name":"host1"
  },
  "source":{
```

```
    "ip":"217.12.12.1"
  }
}

POST /tfm-events/_doc/
{
  "@timestamp": "2021-04-21T16:00:00.000Z",
  "destination":{
    "ip":"193.24.23.13"
  },
  "host":{
    "name":"host2"
  },
  "source":{
    "ip":"217.12.12.2"
  }
}
```

Tras insertar estos dos eventos y esperar a que se ejecute la regla de detección se comprueba que se han creado dos alertas de detección, puesto que la query configurada produce dos resultados.

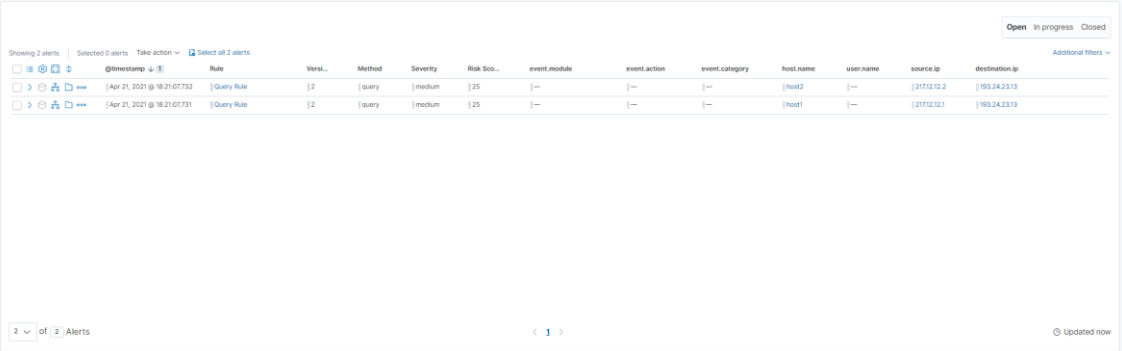

Figura 9. Alertas Custom Query

Además, también se debe haber ejecutado la acción configurada para la alerta de detección y por tanto, se debe haber recibido un mensaje de correo electrónico en la dirección `rpiccio@uoc.edu`.

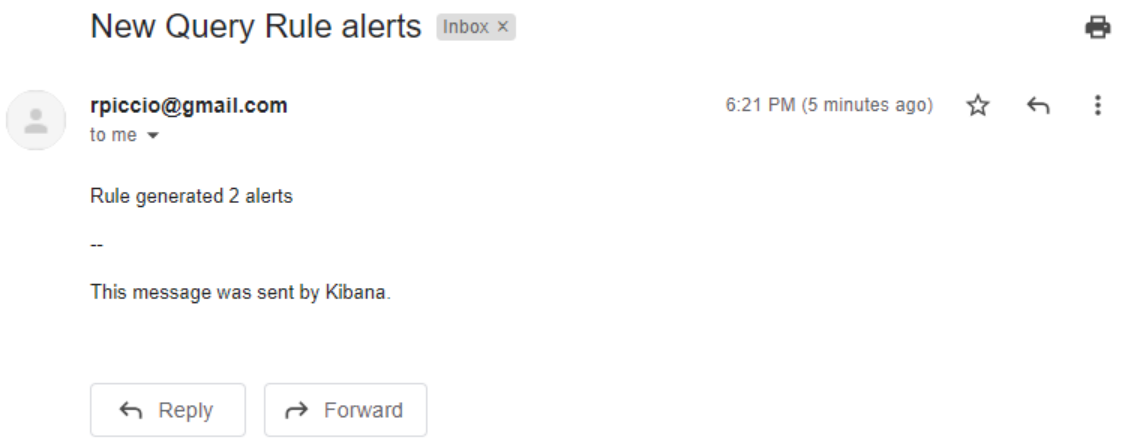

Figura 10. Mensaje de correo electrónico.

Como se puede observar, los *placeholders* que se habían configurado en los campos Subject y Message de la acción de tipo Email se han sustituido por sus valores efectivos (el nombre de la regla y el número de alertas creadas).

Para proba las excepciones se añade a la regla una excepción de tal forma que si el campo `host.name` es igual a `host5` no se genere alerta. Una vez creada esta excepción se añaden dos nuevos eventos al índice `tfm-events`.

```
POST /tfm-events/_doc/
{
  "@timestamp": "2021-04-21T16:00:00.000Z",
  "destination":{
    "ip":"193.24.23.13"
  },
  "host":{
    "name":"host3"
  },
  "source":{
    "ip":"217.12.12.3"
  }
}

POST /tfm-events/_doc/
{
  "@timestamp": "2021-04-21T16:00:00.000Z",
  "destination":{
    "ip":"193.24.23.13"
  },
  "host":{
    "name":"host5"
  },
  "source":{
    "ip":"217.12.12.5"
  }
}
```

Tras ejecutarse la regla se comprueba que el resultado es el esperado. Solamente se ha creado una alerta para el evento con el campo `host.name` con valor `host3`, ya que el otro cumple con lo definido en la excepción.

3.1.2 Reglas Machine Learning

Se crea una alerta de detección cuando un trabajo *machine learning* descubre una anomalía por encima de un umbral configurado. Es necesario que el trabajo machine learning se esté ejecutando.

La documentación sobre *machine learning* en Elastic Stack [27] es detallada y además, no es el objetivo de este Trabajo analizar esas capacidades si no evaluar cómo se pueden usar para detectar posibles incidentes de seguridad.

Por eso partiremos de un entorno de evaluación donde se han cargado los datos de ejemplo de logs de un sitio web y se han creado los trabajos de *machine learning* que acompañan a esos datos de ejemplo. Además esos trabajos se deben añadir al grupo security para poder usarse desde Elastic Security.

Para crear una regla de tipo Machine Learning hay que seleccionar el trabajo que se quiere emplear, el umbral a partir del cual una anomalía producirá una alerta de detección y la plantilla de línea de tiempo que se quiere emplear.

Se creará una alerta de detección llamada Machine Learning Rule, con severidad Medium, con puntuación de riesgo 25 y con la siguiente definición:

a) **Machine Learning Job**: `response_code_rates`

b) **Anomaly Score Threshold**: 50

Esta regla se ejecutará cada minuto con una ventana de 360 horas. Esta ventana se debe a que para los datos de prueba que se cargan existe una anomalía de severidad 77 en los 15 días anteriores a la carga.

Y tras la primera ejecución de la regla de detección se puede comprobar que efectivamente se ha creado una alerta de detección debido a la anomalía antes mencionada de severidad 77.

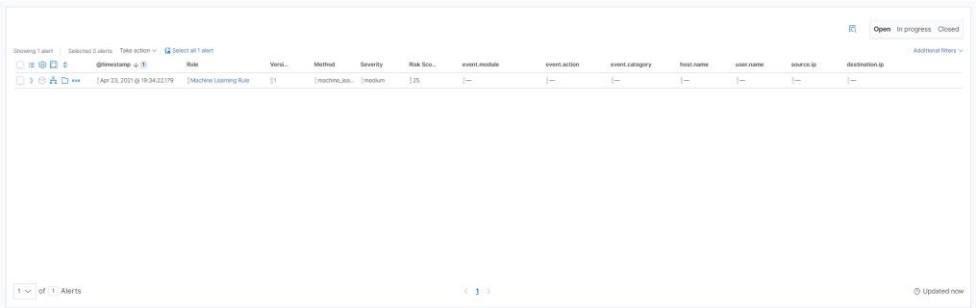

Figura 11. Alertas Machine Learning.

3.1.3 Reglas Threshold

Se crea una alerta de detección cuando el valor de una agregación está por encima de un umbral configurado. Para crear una regla de detección de tipo Threshold es necesario completar los siguientes campos:

a) **Index patterns**: Los patrones de los índices sobre los que se buscarán los documentos.

b) **Custom query**: Los filtros y la consulta que se utilizarán para obtener los documentos.

c) **Group by**: Los campos por los que agrupar y el umbral que deben superar las agrupaciones para que se cree una alerta.

d) **Count**: Un campo de la agrupación que debe superar un determinado umbral para que se cree una alerta.

e) **Timeline template**: La plantilla de línea de tiempo a utilizar de la lista de plantillas de líneas de tiempo disponibles.

Para ilustrar este tipo de reglas, primero se crea en el entorno de evaluación un index template `tfm-events` que aplicará sobre índices de tipo datastream que se llamen `tfm-events`. Este index template define los campos que pueden tener los documentos.

```
PUT /_index_template/tfm-events
{
    "index_patterns":[
        "tfm-events"
    ],
    "data_stream":{
    },
    "template":{
        "mappings":{
            "date_detection":false,
            "dynamic":"strict",
            "dynamic_templates":[
                {
                    "strings_as_keywords":{
                        "match_mapping_type":"string",
                        "mapping":{
                            "type":"keyword"
                        }
                    }
                }
            ],
            "properties":{
                "@timestamp":{
                    "type":"date",
                    "format":"yyyy-MM-dd'T'HH:mm:ss.SSSXXX"
                },
                "destination":{
                    "type":"object",
                    "properties":{
                        "ip":{
                            "type":"ip"
                        }
                    }
                },
                "host":{
                    "type":"object",
                    "properties":{
                        "name":{
                            "type":"keyword"
                        }
                    }
                },
                "source":{
                    "type":"object",
                    "properties":{
                        "ip":{
                            "type":"ip"
                        }
                    }
                }
            }
        }
    }
}
```

Se creará una regla llamada Threshold Rule, con severidad Medium, con puntuación de riesgo 25 y con la siguiente definición:

a) **Index patterns**: `tfm-events`

b) **Filters**: Ninguno.

c) **Custom query**: `destination.ip: 193.24.23.13`

d) **Group by**: `source.ip` con un umbral mayor o igual que 3

e) **Count**: `host.name` con un umbral mayor o igual que 3.

Esta regla de detección se ejecutará cada minuto con una ventana de 12 horas y tendrá además asociada una acción de tipo Email que enviará un correo cada vez que se creen alertas de detección con los siguientes parámetros:

a) **To**: `rpiccio@uoc.edu`

b) **Subject**: `New {{context.rule.name}} alerts`

c) **Message**: `Rule generated {{state.signals_count}} alerts`

A continuación se irán creando documentos en el índice `tfm-events` y se comprobará si tras la ejecución de la regla se han generado nuevas alertas y se ha enviado el mensaje de correo electrónico esperado.

```
POST /tfm-events/_doc/
{
  "@timestamp": "2021-04-21T20:00:00.000Z",
  "destination":{
    "ip":"193.24.23.13"
  },
  "host":{
    "name":"host1"
  },
  "source":{
    "ip":"207.12.12.1"
  }
}
```

Tras insertar este evento y esperar a que se ejecute la regla se comprueba que no se ha creado alerta, ya que aunque la query configurada sí produce resultados, la única agrupación está por debajo del umbral.

```
POST /tfm-events/_doc/
{
  "@timestamp": "2021-04-21T20:00:00.000Z",
  "destination":{
    "ip":"193.24.23.13"
  },
  "host":{
    "name":"host1"
  },
  "source":{
    "ip":"207.12.12.1"
  }
}

POST /tfm-events/_doc/
{
  "@timestamp": "2021-04-21T20:00:00.000Z",
  "destination":{
    "ip":"193.24.23.13"
  },
  "host":{
    "name":"host1"
  },
  "source":{
    "ip":"207.12.12.1"
  }
}
```

Tras insertar estos dos eventos y esperar a que se ejecute la regla se comprueba que no se ha creado alerta, ya que aunque la query configurada sí pro-

duce resultados y la agrupación `source.ip` con valor `197.12.12.1` está por encima del umbral, `host.name` no lo está.

```
POST /tfm-events/_doc/
{
  "@timestamp": "2021-04-21T20:00:00.000Z",
  "destination":{
    "ip":"193.24.23.13"
  },
  "host":{
    "name":"host2"
  },
  "source":{
    "ip":"207.12.12.1"
  }
}

POST /tfm-events/_doc/
{
  "@timestamp": "2021-04-21T20:00:00.000Z",
  "destination":{
    "ip":"193.24.23.13"
  },
  "host":{
    "name":"host3"
  },
  "source":{
    "ip":"207.12.12.1"
  }
}
```

Tras insertar estos dos eventos y esperar a que se ejecute la regla se comprueba que se ha creado una alerta, ya que la query configurada sí produce resultados, la agrupación `source.ip` con valor `197.12.12.1` está por encima del umbral (3) y `host.name` también lo está (3).

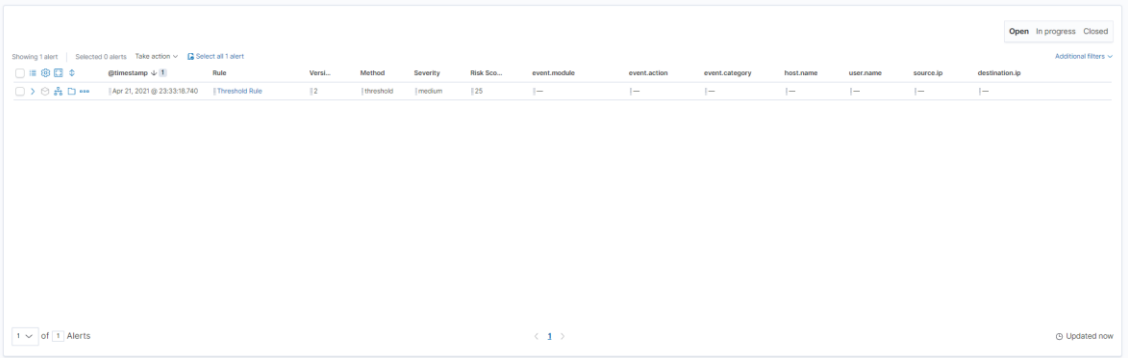

Figura 12. Alerta Threshold

Además, también se debe haber ejecutado la acción configurada para la alerta y por tanto, se debe haber recibido un mensaje de correo electrónico en la dirección `rpiccio@uoc.edu`.

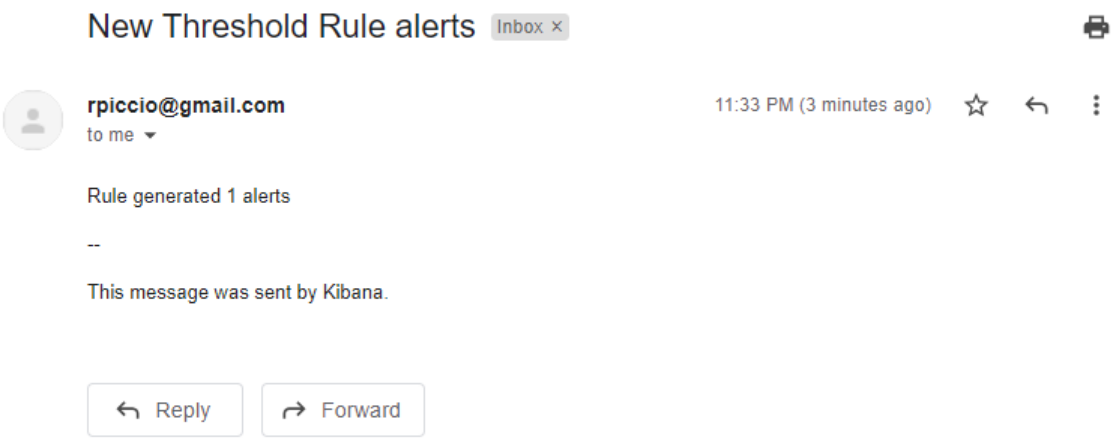

Figura 13. Mensaje de correo electrónico.

Como se puede observar, los *placeholders* que se habían configurado en los campos Subject y Message de la acción de tipo Email se han sustituido por sus valores efectivos (el nombre de la regla y el número de alertas creadas).

3.1.4 Reglas Event Correlation

Se crea una alerta cuando se encuentran documentos que cumplen con una expresión EQL. EQL son las siglas de *Event Query Language*, un lenguaje que permite modelar de forma intuitiva consultas complejas sobre series temporales de eventos.

Aunque a priori el lenguaje EQL se puede utilizar con cualquier serie temporal de eventos, está diseñado especialmente para ser usado en escenarios de detección de amenazas. Y al contrario que otros lenguajes de consulta permite expresar relaciones entre eventos.

Una consulta EQL requiere una categoría de evento y una condición, conectados mediante la palabra clave `where`. Por ejemplo, si se quieren obtener eventos con categoría de evento `process` y campo `process.name` con valor `svchost.exe` la expresión EQL sería:

```
process where process.name == "svchost.exe"
```

Se pueden expresar secuencias para describir una serie ordenada de eventos. Cada elemento de la secuencia es una consulta EQL. Los eventos se ordenan cronológicamente con el evento más reciente en último lugar (ascendente).

```
sequence
  [ event_category_1 where condition_1 ]
  [ event_category_2 where condition_2 ]
```

Estas y otras características del lenguaje como expresar condiciones complejas, usar ventanas temporales mediante el uso de la directiva `with maxspan` o requerir campos comunes entre eventos mediante el uso de la directiva `by` se pueden consultar en [24].

Para crear una regla de detección de tipo Event Correlation es necesario completar los siguientes campos:

a) **Index patterns**: Los patrones de los índices sobre los que se buscarán los documentos.

b) **EQL query**: La consulta que se utiliza para obtener los eventos expresada en lenguaje EQL.

c) **Timeline template**: La plantilla de línea de tiempo a utilizar de la lista de plantillas de líneas de tiempo disponibles.

Para ilustrar este tipo de reglas, primero se crea en el entorno de evaluación un index template `tfm-events` que aplicará sobre índices de tipo datastream que se llamen `tfm-events`. Este index template define los campos que pueden tener los documentos.

```
PUT /_index_template/tfm-events
{
    "index_patterns":[
        "tfm-events"
    ],
    "data_stream":{
    },
    "template":{
        "mappings":{
            "date_detection":false,
            "dynamic":"strict",
            "dynamic_templates":[
                {
                    "strings_as_keywords":{
                        "match_mapping_type":"string",
                        "mapping":{
                            "type":"keyword"
                        }
                    }
                }
            ],
            "properties":{
                "@timestamp":{
                    "type":"date",
                    "format":"yyyy-MM-dd'T'HH:mm:ss.SSSXXX"
                },
                "destination":{
                    "type":"object",
                    "properties":{
                        "ip":{
                            "type":"ip"
                        }
                    }
                },
                "event":{
                    "type":"object",
                    "properties":{
                        "category":{
                            "type":"keyword"
                        },
                        "kind":{
                            "type":"keyword"
                        }
                    }
                },
                "source":{
                    "type":"object",
                    "properties":{
                        "ip":{
```

```
                        "type":"ip"
                    }
                }
            }
        }
    }
  }
}
```

Se creará una regla de detección llamada Event Correlation Rule, de severidad Medium, con puntuación de riesgo 25 y con la siguiente definición:

a) **Index patterns**: `tfm-events`

b) **EQL query**

```
sequence by source.ip
[ network where destination.ip == "193.24.23.12" ]
[ network where destination.ip == "213.24.23.12" ]
```

Esta regla de detección se ejecutará cada minuto con una ventana de 12 horas y tendrá además asociada una acción de tipo Email que enviará un correo cada vez que se creen alertas de detección con los siguientes parámetros:

a) **To**: `rpiccio@uoc.edu`

b) **Subject**: `New {{context.rule.name}} alerts`

c) **Message**: `Rule generated {{state.signals_count}} alerts`

A continuación se irán creando documentos en el índice `tfm-events` y se comprobará si tras la ejecución de la regla se han generado nuevas alertas y se ha enviado el mensaje de correo electrónico esperado.

```
POST /tfm-events/_doc/
{
  "@timestamp": "2021-04-22T19:30:00.000Z",
  "destination":{
    "ip":"193.24.23.12"
  },
  "event":{
    "category":"network",
    "kind":"event"
  },
  "source":{
    "ip":"217.12.12.1"
  }
}
```

Tras insertar este evento y esperar a que se ejecute la regla se comprueba que no se ha creado alerta, ya que la query espera una secuencia de eventos y de momento solo hay uno en el índice.

```
POST /tfm-events/_doc/
{
  "@timestamp": "2021-04-22T19:30:01.000Z",
  "destination":{
    "ip":"203.24.23.12"
  },
  "event":{
    "category":"network",
    "kind":"event"
```

```
  },
  "source":{
    "ip":"217.12.12.1"
  }
}
```

Tras insertar este evento y esperar a que se ejecute la regla se comprueba que no se ha creado alerta, ya que la query espera una secuencia de eventos con el campo `destination.ip` con otros valores.

```
POST /tfm-events/_doc/
{
  "@timestamp": "2021-04-22T19:30:02.000Z",
  "destination":{
    "ip":"213.24.23.12"
  },
  "event":{
    "category":"network",
    "kind":"event"
  },
  "source":{
    "ip":"217.12.12.2"
  }
}
```

Tras insertar este evento y esperar a que se ejecute la regla se comprueba que no se ha creado alerta, ya que la query espera una secuencia de eventos con el mismo valor en el campo `source.ip`.

```
POST /tfm-events/_doc/
{
  "@timestamp": "2021-04-22T19:30:03.000Z",
  "destination":{
    "ip":"213.24.23.12"
  },
  "event":{
    "category":"network",
    "kind":"event"
  },
  "source":{
    "ip":"217.12.12.1"
  }
}
```

Tras insertar este evento y esperar a que se ejecute la regla se comprueba que se ha creado una alerta. La secuencia de eventos es la esperada y además tienen el mismo valor en el campo `source.ip`.

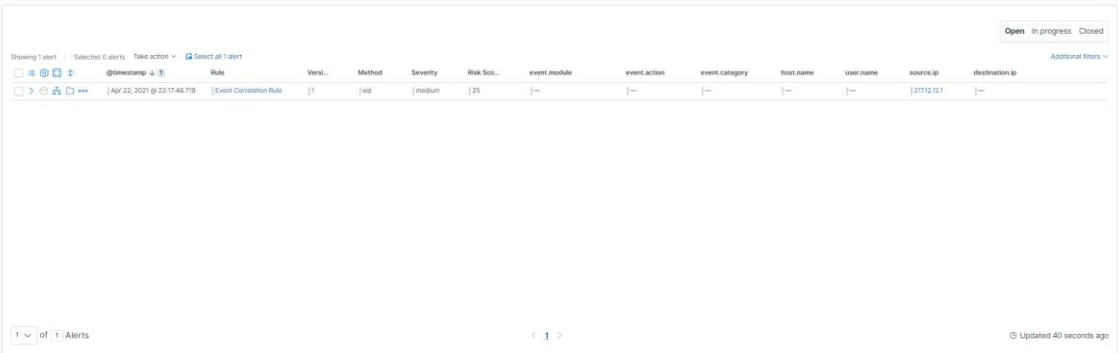

Figura 14. Alerta de tipo Indicator Match.

Además, también se debe haber ejecutado la acción configurada para la alerta y por tanto, se debe haber recibido un mensaje de correo electrónico en la dirección `rpiccio@uoc.edu`.

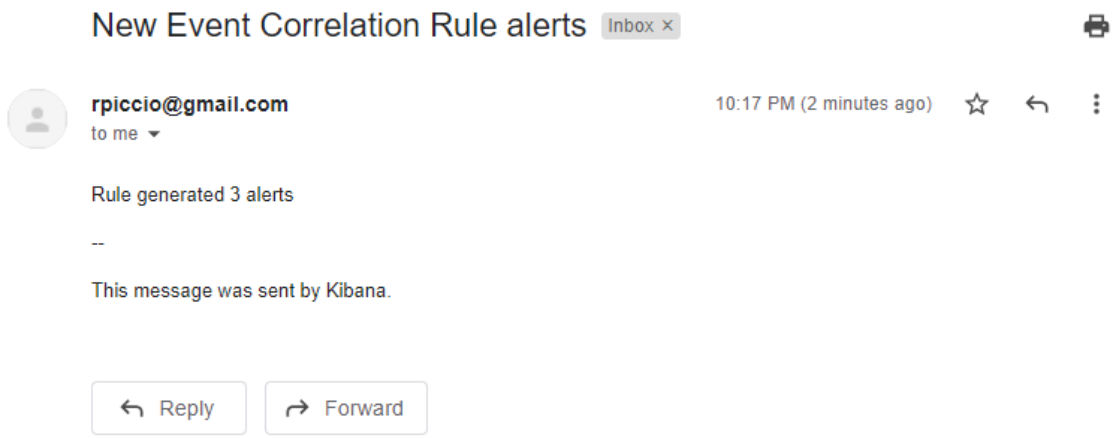

Figura 15. Mensaje de correo electrónico.

Como se puede observar, los *placeholders* que se habían configurado en los campos Subject y Message de la acción de tipo Email se han sustituido por sus valores efectivos (el nombre de la regla y el número de alertas creadas).

3.1.5 Reglas Indicator Match

Se crea una alerta cuando se encuentran documentos que tienen campos con ciertos valores que se obtienen a su vez de otra consulta (*indicator*). Para crear una regla Indicator Match es necesario completar los siguientes campos:

a) **Index patterns**: Los patrones de los índices sobre los que se buscarán los documentos.

b) **Custom query**: Los filtros y la consulta que se utilizarán para obtener los documentos.

c) **Indicator index patterns**: Los patrones de los índices de los que se obtendrán los valores de referencia.

d) **Indicator index query**: Los filtros y la consulta que se utilizan para obtener los valores de referencia.

e) **Indicator mapping**: Cómo combinar y comparar los valores de los eventos con los valores de referencia.

f) **Timeline template**: La plantilla de línea de tiempo a utilizar de la lista de plantillas de líneas de tiempo disponibles.

Para ilustrar este tipo de reglas, primero se crea en el entorno de evaluación un index template `tfm-events` que aplicará sobre índices de tipo datastream que se llamen `tfm-events`. Este index template define los campos que pueden tener los documentos.

```
PUT /_index_template/tfm-events
{
    "index_patterns":[
        "tfm-events"
    ],
    "data_stream":{
    },
    "template":{
        "mappings":{
            "date_detection":false,
            "dynamic":"strict",
            "dynamic_templates":[
                {
                    "strings_as_keywords":{
                        "match_mapping_type":"string",
                        "mapping":{
                            "type":"keyword"
                        }
                    }
                }
            ],
            "properties":{
                "@timestamp":{
                    "type":"date",
                    "format":"yyyy-MM-dd'T'HH:mm:ss.SSSXXX"
                },
                "destination":{
                    "type":"object",
                    "properties":{
                        "ip":{
                            "type":"ip"
                        }
                    }
                },
                "host":{
                    "type":"object",
                    "properties":{
                        "name":{
                            "type":"keyword"
                        }
                    }
                },
                "source":{
                    "type":"object",
                    "properties":{
                        "ip":{
                            "type":"ip"
                        }
                    }
                }
            }
        }
    }
}
```

Se creará también un index template `tfm-indicators` que aplicará sobre índices de tipo datastream que se llamen `tfm-indicators`. Este index template define los campos que pueden tener los documentos.

```
PUT /_index_template/tfm-indicators
{
    "index_patterns":[
        "tfm-indicators"
    ],
    "data_stream":{
    },
    "template":{
        "mappings":{
            "date_detection":false,
            "dynamic":"strict",
            "dynamic_templates":[
```

```
                {
                    "strings_as_keywords":{
                        "match_mapping_type":"string",
                        "mapping":{
                            "type":"keyword"
                        }
                    }
                }
            ],
            "properties":{
                "@timestamp":{
                    "type":"date",
                    "format":"yyyy-MM-dd'T'HH:mm:ss.SSSXXX"
                },
                "threat":{
                    "type":"object",
                    "properties":{
                        "indicator":{
                            "type":"object",
                            "properties":{
                                "ip":{
                                    "type":"ip"
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```

La organización debe tener un proceso que irá añadiendo documentos al índice `tfm-indicators` en función de sus fuentes de inteligencia. Existen integraciones de Elastic [28] con fuentes como abuse.ch [29] o AlienVault OTX [30] pero para el ejemplo se crearán un par de documentos de forma manual:

```
POST /tfm-indicators/_doc/
{
  "@timestamp": "2021-04-01T00:00:00.000Z",
  "threat":{
    "indicator":{
      "ip": "193.24.23.13"
    }
  }
}

POST /tfm-indicators/_doc/
{
  "@timestamp": "2021-04-01T00:00:00.000Z",
  "threat":{
    "indicator":{
      "ip": "193.24.23.14"
    }
  }
}
```

Se creará una regla de detección llamada Indicator Match Rule, con severidad Medium, con puntuación de riesgo 25 y con la siguiente definición:

a) **Index patterns**: `tfm-events`

b) **Filters**: `host.name: host1`

c) **Custom query**: `destination.ip: *`

d) **Indicator index patterns**: `tfm-indicators`

e) **Indicator index query**: `threat.indicator.ip: *`

f) **Indicator mapping**: `destination.ip` MATCHES `threat.indicator.ip`

Esta regla de detección se ejecutará cada minuto con una ventana de 12 horas y tendrá además asociada una acción de tipo Email que enviará un correo cada vez que se creen alertas de detección con los siguientes parámetros:

d) **To**: `rpiccio@uoc.edu`

e) **Subject**: `New {{context.rule.name}} alerts`

f) **Message**: `Rule generated {{state.signals_count}} alerts`

A continuación se irán creando documentos en el índice `tfm-events` y se comprobará si tras la ejecución de la regla se han generado nuevas alertas y se ha enviado el mensaje de correo electrónico esperado.

```
POST /tfm-events/_doc/
{
  "@timestamp": "2021-04-20T22:00:00.000Z",
  "destination":{
    "ip":"193.24.23.12"
  },
  "host":{
    "name":"host1"
  },
  "source":{
    "ip":"217.12.12.1"
  }
}
```

Tras insertar este evento y esperar a que se ejecute la regla se comprueba que no se ha creado alerta, ya que aunque el evento cumple con el filtro configurado el campo **`destination.ip`** no tiene un valor del índice de referencia.

```
POST /tfm-events/_doc/
{
  "@timestamp": "2021-04-20T22:00:00.000Z",
  "destination":{
    "ip":"193.24.23.13"
  },
  "host":{
    "name":"host2"
  },
  "source":{
    "ip":"217.12.12.2"
  }
}
```

Tras insertar este evento y esperar a que se ejecute la regla se comprueba que no se ha creado alerta, ya que el evento no cumple con el filtro configurado aunque el campo **`destination.ip`** tiene un valor del índice de referencia.

```
POST /tfm-events/_doc/
{
  "@timestamp": "2021-04-20T22:00:00.000Z",
  "destination":{
    "ip":"193.24.23.13"
  },
  "host":{
    "name":"host1"
```

```
  },
  "source":{
    "ip":"217.12.12.1"
  }
}
```

Tras insertar este evento y esperar a que se ejecute la regla se comprueba que se ha creado una alerta. El evento cumple el filtro configurado y además el campo **`destination.ip`** tiene un valor del índice de referencia.

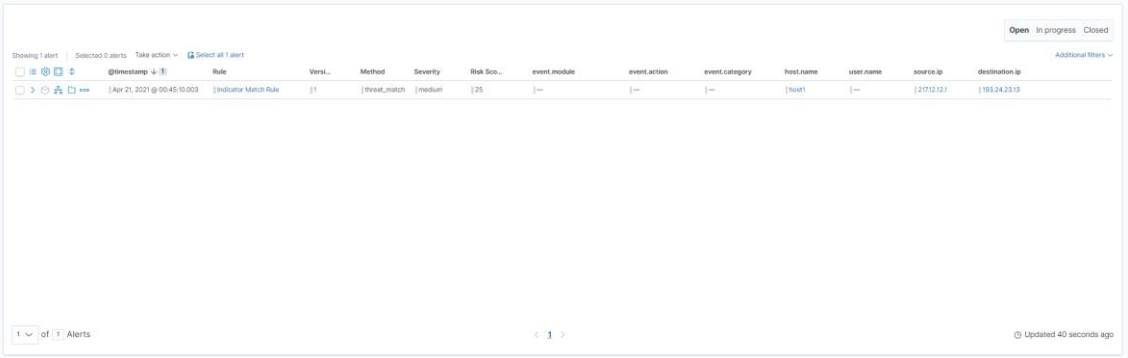

Figura 16. Alerta de tipo Indicator Match.

Además, también se debe haber ejecutado la acción configurada para la alerta y por tanto, se debe haber recibido un mensaje de correo electrónico en la dirección `rpiccio@uoc.edu`.

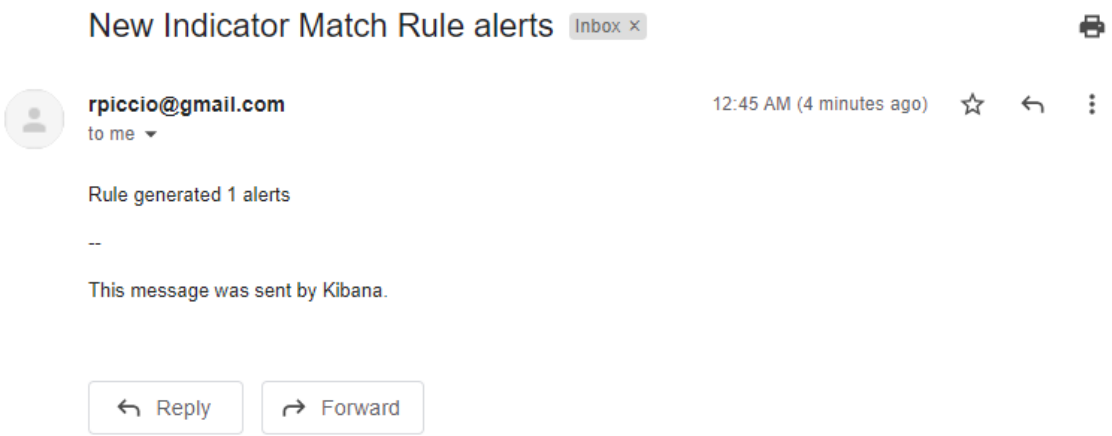

Figura 17. Mensaje de correo electrónico.

Como se puede observar, los *placeholders* que se habían configurado en los campos Subject y Message de la acción de tipo Email se han sustituido por sus valores efectivos (el nombre de la regla y el número de alertas creadas).

Para proba las excepciones se añade a la regla una excepción de tal forma que si el campo `source.ip` es igual a `207.12.12.1` no se genere alerta. Una vez creada esta excepción se añade un nuevo eventos al índice `tfm-events`.

```
POST /tfm-events/_doc/
{
  "@timestamp": "2021-04-20T22:00:00.000Z",
```

```
  "destination":{
    "ip":"193.24.23.13"
  },
  "host":{
    "name":"host1"
  },
  "source":{
    "ip":"207.12.12.1"
  }
}
```

Tras ejecutarse la regla se comprueba que el resultado es el esperado. Aunque el evento cumple el filtro configurado y el campo `destination.ip` tiene un valor del índice de referencia, como el campo `source.ip` cumple con la excepción definida no se crea alerta.

3.2 Análisis

Las capacidades de análisis que ofrece Elastic Security para ayudar a los analistas a detectar, investigar y gestionar posibles incidentes de seguridad se basan principalmente en dos módulos, el módulo de Líneas de Tiempo (Timelines) y el módulo de Casos (Cases).

3.2.1 Líneas de Tiempo

El módulo de Líneas de Tiempo ofrece espacios de trabajo diseñados específicamente para que los analistas puedan investigar situaciones que consideren sospechosas, a partir de eventos o alertas que aparezcan en Elastic Security o asociadas a actividades más genéricas de *threat hunting*.

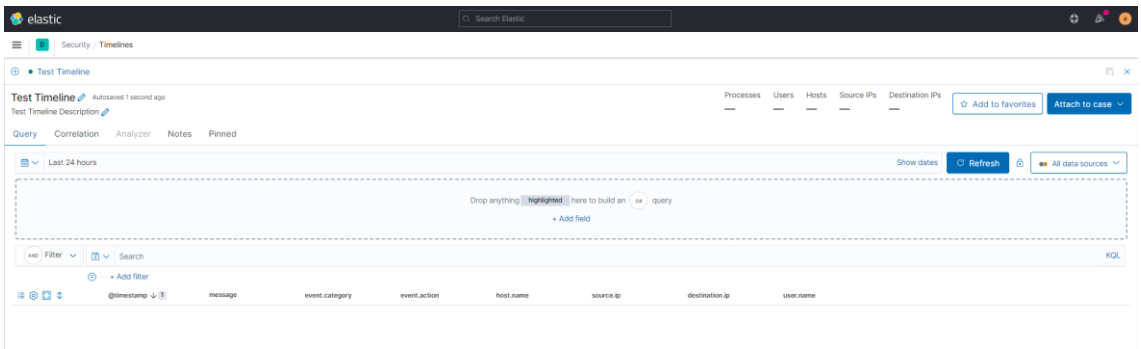

Figura 18. Línea de Tiempo.

Una línea de tiempo estará en modo borrador hasta que la asignemos un nombre distinto del que Elastic Security la asigna por defecto (Untitled timeline). A partir de ese momento estará disponible en la lista de líneas de tiempo y se podrá volver a ella siempre que sea necesario.

En una línea de tiempo hay cinco pestañas (aunque Analyzer está orientada a Endpoint Security y por tanto de momento no se va a evaluar). La primera de ellas, llamada Query, permite construir la consulta KQL necesaria para que en la línea de tiempo aparezcan todos los documentos que le puedan interesen al analista para la investigación que esté realizando en ese momento.

Por ejemplo, una consulta que muestre todos los eventos de las últimas 24 horas que tengan el valor `142.250.183.3` en los campos `destination.ip` o `source.ip` y que en el campo `host.name` tengan el valor `tfm-debian`:

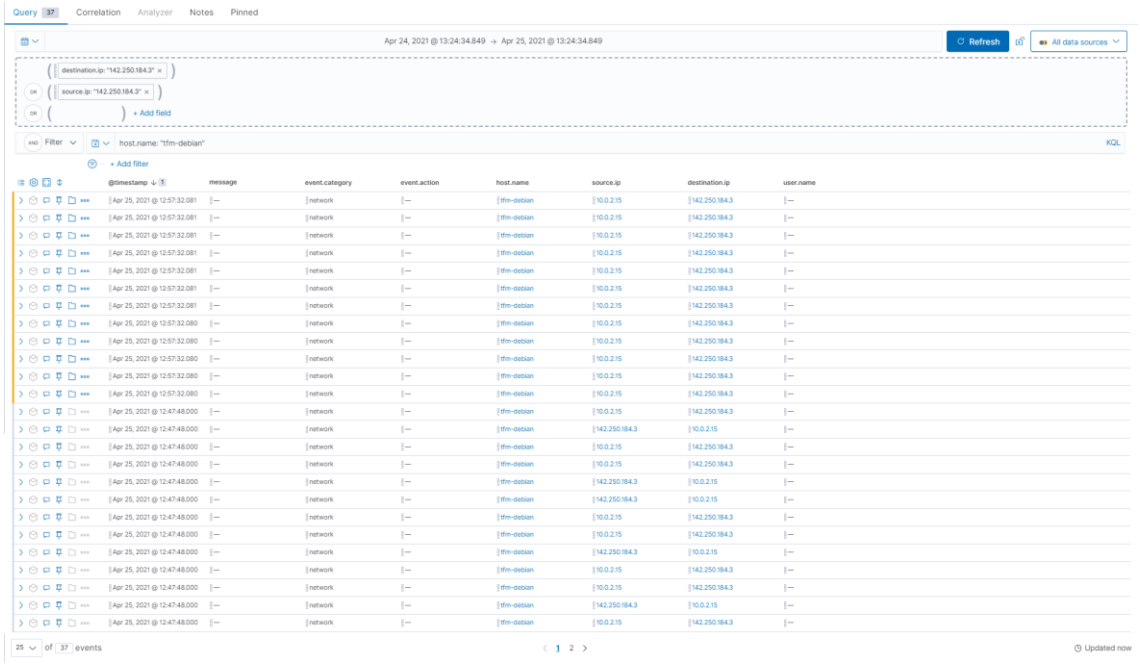

Figura 19. Consulta KQL en Línea de Tiempo.

La siguiente pestaña, llamada Correlation, es muy similar a Query pero en lugar de utilizar consultas KQL permite trabajar con consultas EQL y por tanto aprovechar todas las capacidades de correlación de eventos que ofrece dicho lenguaje (y de ahí el nombre de la pestaña).

Las columnas que se muestran en ambas pestañas son configurables. Por defecto se muestran una serie de campos que habitualmente son útiles en una investigación pero se puede elegir cualquier combinación de campos ECS [31] y en el orden que más interese al analista.

En el caso de consultas EQL el orden en el que aparecen los eventos en la tabla viene implícito por el propio funcionamiento del lenguaje de consulta pero para consultas KQL además de elegir los campos de la tabla también es posible establecer los criterios de ordenación de los documentos.

Por cada documento de la tabla tenemos tres acciones relevantes. Por un lado, podemos ver el detalle del documento en formato tabla y con posibilidad de filtrar por nombre o valor de campo, el detalle del documento en formato JSON y, si se trata de una alerta de detección, un resumen de la alerta con sus campos principales.

También se pueden fijar documentos. Estos documentos pasarán a estar accesibles en la pestaña llamada Pinned, con independencia de que se cambie la query KQL o EQL que se estuviera empleando y el documento en cuestión ya no apareciese en las otras pestañas.

Por último, también es posible añadir notas a todos aquellos documentos que los analistas consideren relevantes para la investigación. Esto tiene tres implicaciones: el documento queda fijado, las notas se muestran siempre que se muestra el documento y además en la pestaña Notes se muestran todas las notas de todos los documentos.

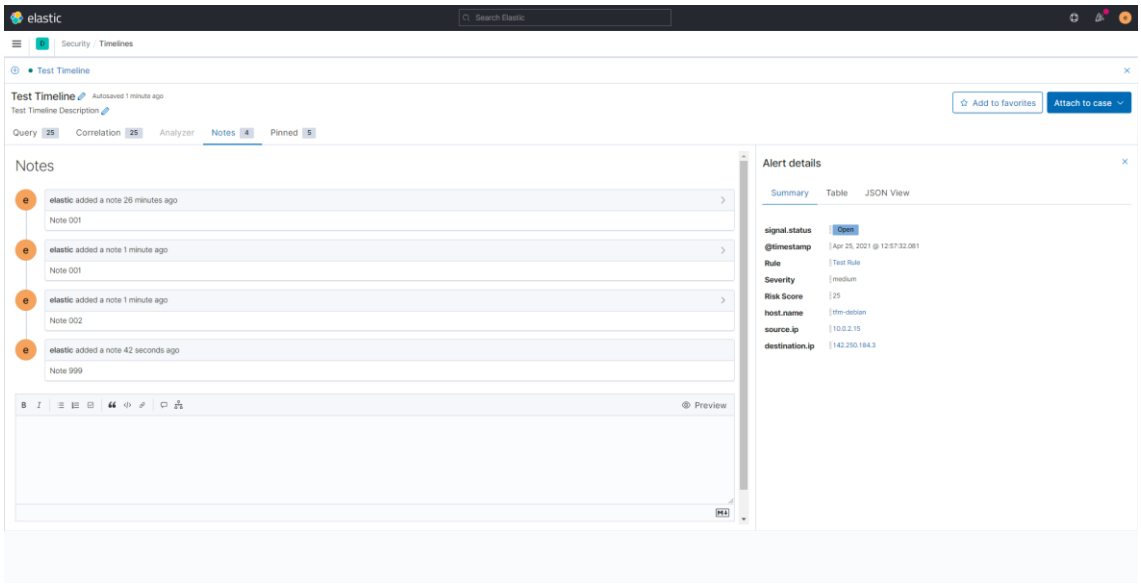

Figura 20. Notas en Línea de Tiempo.

En la pestaña Notas además es posible acceder a los detalles del documento para el que se ha añadido la nota (tanto si es una alerta de detección como si es un evento) y añadir notas generales de la línea de tiempo, notas que no estarán asociadas a ningún documento.

3.2.2 Casos

Elastic Security ofrece un módulo de Casos para poder gestionar posibles incidentes de seguridad. Aunque se puede gestionar cualquier incidente de seguridad lógicamente el módulo está pensado fundamentalmente para gestionar los incidentes detectados por la propia aplicación.

Además es posible sincronizar los casos con herramientas externas como ServiceNow [32], Jira Software [33], Jira Service Management [34] o IBM Resilient [35] siempre que se disponga de la licencia adecuada. Las integraciones con Jira requieren una licencia *gold* mientras que el resto de integraciones requieren una licencia *platinum*.

Se pueden asociar casos con líneas de tiempo y con alertas de detección. Si se asocia un caso con alertas de detección, además se puede vincular el estado del caso con el estado de las alertas de detección.

Accediendo al módulo de Casos lo primero que se presenta es una vista general que permite obtener de forma rápida una instantánea de los casos en forma de tabla que incluye el nombre del caso, quien lo ha creado, su estado, sus etiquetas o si tiene comentarios y sobre la que se pueden aplicar filtros.

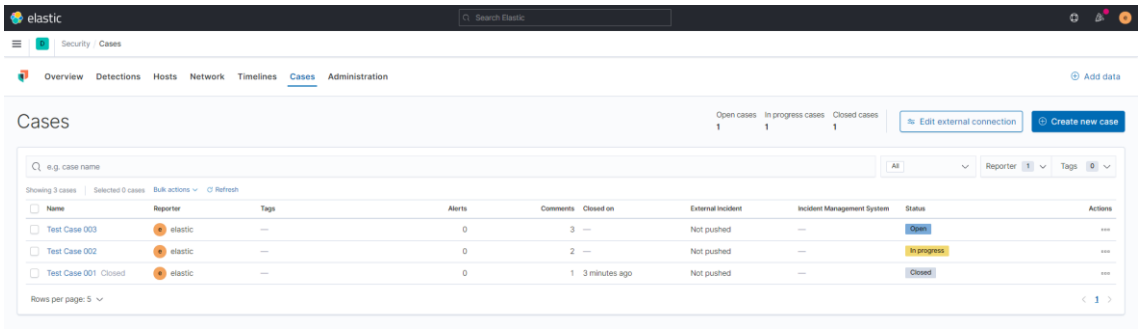

Figura 21. Casos.

Seleccionando el nombre de un caso se puede acceder a la vista de detalle. Se trata de una vista bastante sencilla en donde se puede ver tanto la descripción del caso como los comentarios que han ido introduciendo los analistas.

Estos comentarios pueden ser de tres tipos: comentarios normales en donde se pueden indicar aspectos relevantes de la gestión del incidente, alertas que se han asociado al caso o líneas de tiempo que se han asociado al caso:

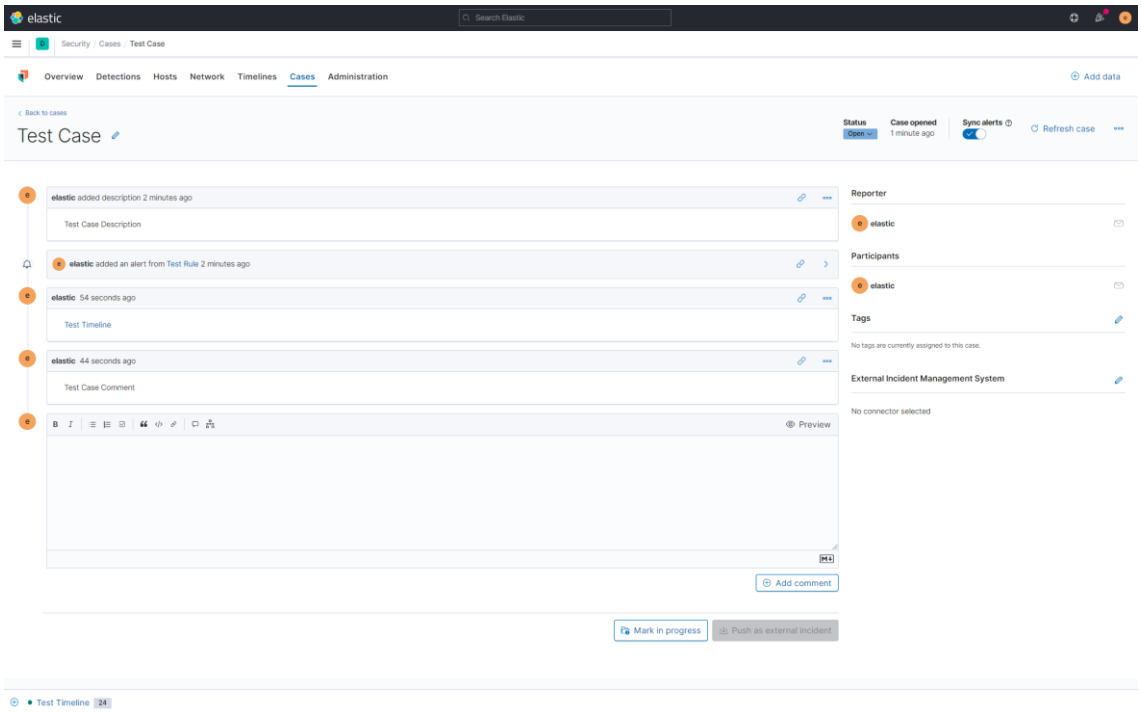

Figura 22. Detalle de un Caso.

Por último, desde la vista de detalle también se puede actualizar el estado del caso, que puede ser: Open para casos que necesitan ser investigados, In Progress para casos que están siendo activamente investigados o Closed para casos ya resueltos.

Este último estado Closed no es final ya que, si es necesario, es posible reabrir los casos. Se comprueba también en el entorno de evaluación que si el estado del caso está vinculado al estado de las alertas de detección, el estado de estas también cambia al actualizar el estado del caso.

3.3 Integración

Como ya se ha visto brevemente en los apartados anteriores Elastic Security dispone de capacidades de integración con sistemas externos tanto en el módulo de Detecciones a través de la ejecución de acciones al crearse nuevas alertas como en el módulo de Casos, en donde es posible sincronizar los casos de Elastic Security con soluciones de terceros como ServiceNow o Jira.

Sin embargo, todas estas capacidades, incluso las que son genéricas como las acciones de propósito general que permiten enviar un correo electrónico o invocar una URL externa (webhook) cuando se detecta una alerta requieren una licencia comercial oro o platino.

Como Elastic Security ofrece también casi todas sus capacidades en forma de API parece interesante analizar en este Trabajo si para aquellas organizaciones donde adquirir una licencia comercial no sea posible existe alguna alternativa para la integración con terceros mediante el uso de dichas APIs.

3.3.1 Signals

Dentro del API que ofrece el módulo de Detecciones uno de los endpoints disponibles es el de Signals, que permite obtener y agregar alertas y actualizar su estado.

Para obtener alertas de detección se debe hacer una llamada HTTP usando el método POST a la siguiente URL:

```
https://kibana.tfm.kanoon.eu:5601/api/detection_engine/signals/search
```

El cuerpo de la llamada se debe construir siguiendo Query DSL [36]. Por ejemplo, se pueden solicitar todas las alertas:

```
curl --data-ascii @query.json \
     --header "content-type: application/json" \
     --header "kbn-xsrf: true" \
     --request POST \
     --user user:password \
     --verbose https://kibana.tfm.kanoon.eu:5601/api/detection_engine/signals/search ; echo
```

siendo el contenido del fichero `query.json` el siguiente:

```
{
    "query": {
        "match_all": {
        }
    }
}
```

Y se obtendrán resultados similares a estos:

```
{
   "took":19,
   "timed_out":false,
   "_shards":{
      "total":2,
      "successful":2,
      "skipped":0,
```

```
      "failed":0
   },
   "hits":{
      "total":{
         "value":1,
         "relation":"eq"
      },
      "max_score":1,
      "hits":[
         {
            "_index":".siem-signals-default-000002",
            "_type":"_doc",
            "_id":"be784e1c42e0b3d67f0758a7a2cc78552afad784406d30e60a8e912a6a717551",
            "_score":1,
            "_source":{
               "@timestamp":"2021-05-02T12:38:05.445Z",
               . . .
               "signal":{
                  . . .
                  "status":"in-progress"
                  . . .
               }
            }
         }
      ]
   }
}
```


Como se puede observar, es sencillo obtener las alertas de detección y cada alerta de detección presente en la respuesta tiene un campo `@timestamp` con la fecha de creación de la alerta.

Por tanto, es factible desarrollar un proceso que haga peticiones periódicas de búsqueda de alertas, obtenga todas aquellas creadas desde la fecha de la última consulta e implemente las acciones que se quieran ejecutar por cada nueva alerta, como por ejemplo, enviar un correo electrónico.

3.3.2 Cases

El API del módulo de Casos ofrece varios endpoints que permiten acceder de forma programática a las mismas funcionalidades disponibles desde la interfaz de usuario de Kibana.

Permite obtener un listado de casos a través del endpoint Find Cases, basado en el API Find Objects de Kibana [37] para obtener saved objects. Bastará con hacer una llamada HTTP usando el método GET a la siguiente URL:

```
https://kibana.tfm.kanoon.eu:5601/api/cases/_find
```

Mediante parámetros de la URL es posible definir los criterios de ordenación y de paginación que se quieren emplear además de poder filtrar los casos por ciertos campos como `status`, `reporters` o `tags`. Los resultados serán similares a estos:


```
{
   "page":1,
   "per_page":20,
   "total":1,
   "cases":[
      {
         "id":"be4effb0-ab4a-11eb-aa5f-19242ea4ba45",
         "version":"WzY2OSw0XQ==",
```

```
        "comments":[

        ],
        "totalComment":1,
        "totalAlerts":1,
        "type":"individual",
        "title":"Test Case",
        "tags":[

        ],
        "description":"Test Case",
        "connector":{
           "id":"none",
           "name":"none",
           "type":".none",
           "fields":null
        },
        "settings":{
           "syncAlerts":true
        },
        "closed_at":null,
        "closed_by":null,
        "created_at":"2021-05-02T13:31:47.362Z",
        "created_by":{
           "email":null,
           "full_name":null,
           "username":"elastic"
        },
        "external_service":null,
        "status":"open",
        "updated_at":"2021-05-02T13:31:49.316Z",
        "updated_by":{
           "full_name":null,
           "email":null,
           "username":"elastic"
        }
     }
   ],
   "count_open_cases":1,
   "count_in_progress_cases":0,
   "count_closed_cases":0
}
```


Existen también endpoints para obtener todos los detalles de un caso concreto, como los comentarios o las alertas asociadas, así como para actualizar campos y comentarios.

Por lo tanto, a priori también parece posible implementar un proceso externo que realice consultas periódicas de los casos existentes en Elastic Security, obtenga sus detalles y los mantenga sincronizados con un sistema externo, siempre que el sistema externo ofrezca capacidades similares si se quiere que la sincronización sea bidireccional.

Sin embargo, la documentación apunta a que el API de consulta no está pensando para recorrer un número alto de casos y no se ha encontrado una forma de filtrar los casos por fecha de actualización (aunque se ha preguntado en los foros del producto).

Si finalmente no fuese posible y el API de consulta se fuese degradando habría que buscar alguna solución de compromiso, como por ejemplo sólo consultar los casos no cerrados (suponiendo que su número debería ser relativamente pequeño) asumiendo que se podría llegar a perder algún caso si se abriese y cerrase durante el intervalo entre consultas.

4. Evaluación Endpoint Security

Además de la funcionalidad de SIEM, Elastic Security permite también proteger los endpoints de la organización mediante Endpoint Security. Endpoint Security se integra con Elastic Agent, que a su vez se puede desplegar a través de Fleet [38], lo cual simplifica en gran medida su gestión, sobre todo en organizaciones cuyo número de endpoints empieza a ser elevado.

4.1 Fleet

Fleet permite gestionar las integraciones con Elastic Stack de un número significativo de servicios y plataformas, así como gestionar los despliegues de Elastic Agent. Para poder evaluar Fleet lo primero es configurarlo adecuadamente accediendo a la opción Settings desde cualquier página la aplicación.

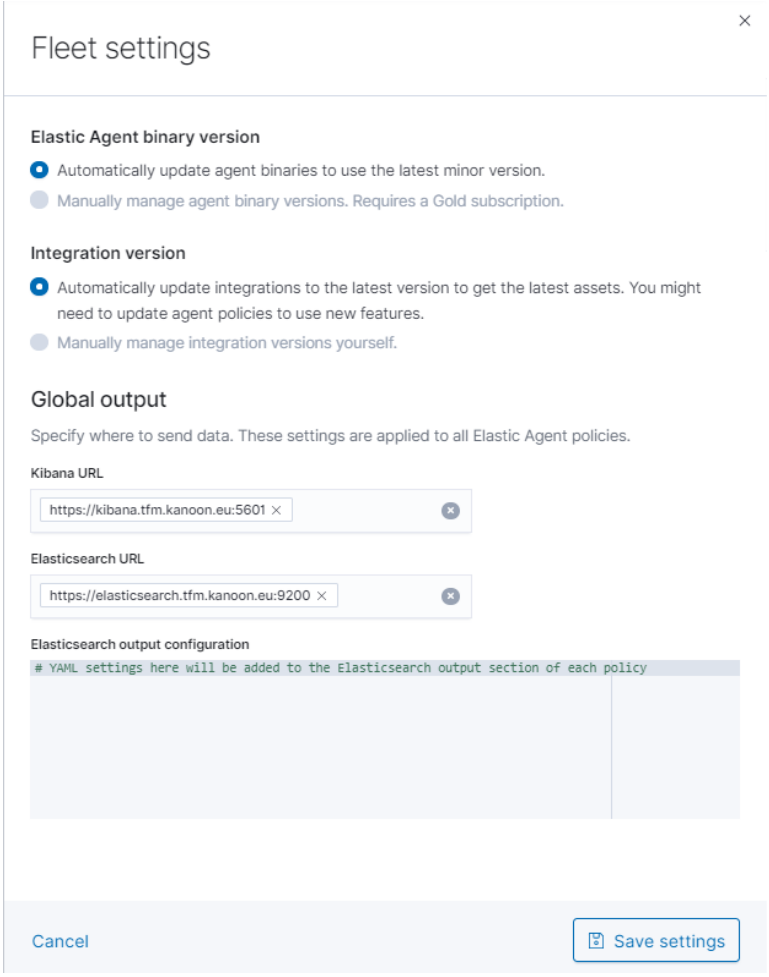

Figura 23. Configuración de Fleet.

Los dos parámetros más relevantes de la configuración son las URLs externas de Kibana y de Elasticsearch. Estas URLs deben ser accesibles desde todos los equipos donde se vaya a instalar Elastic Agent. Si no son accesibles fallará tanto el despliegue de nuevas configuraciones e integraciones como el envío de métricas y logs.

En la pestaña Integrations se pueden instalar y configurar las integraciones que vaya a querer usar la organización. Hay dos integraciones que vienen instaladas por defecto: System, que permite recoger tanto logs como métricas básicas de los equipos y Endpoint Security.

Existen integraciones para recoger información de clouds públicas como Azure, AWS o Google Cloud, de servidores web como Apache o NGINX, de bases de datos como MongoDB, Redis o MySQL o de productos de red y de seguridad como F5 BigIP, cortafuegos Check Point o soluciones de Fortinet.

Cada integración define una serie de assets tanto de Elasticsearch como de Kibana que es necesario instalar antes de poder utilizar la integración. Por lo general se trata de visualizaciones, dashboards o pipelines de ingestión para extraer campos estructurados de logs.

El segundo concepto relevante para Fleet son las políticas, que se gestionan desde la pestaña de Policies. Desde esta pestaña es posible crear nuevas políticas así como visualizar una lista a modo de resumen de todas las políticas definidas. Desde esta lista es posible acceder a cualquier política para ver el detalle y para modificar lo que sea necesario.

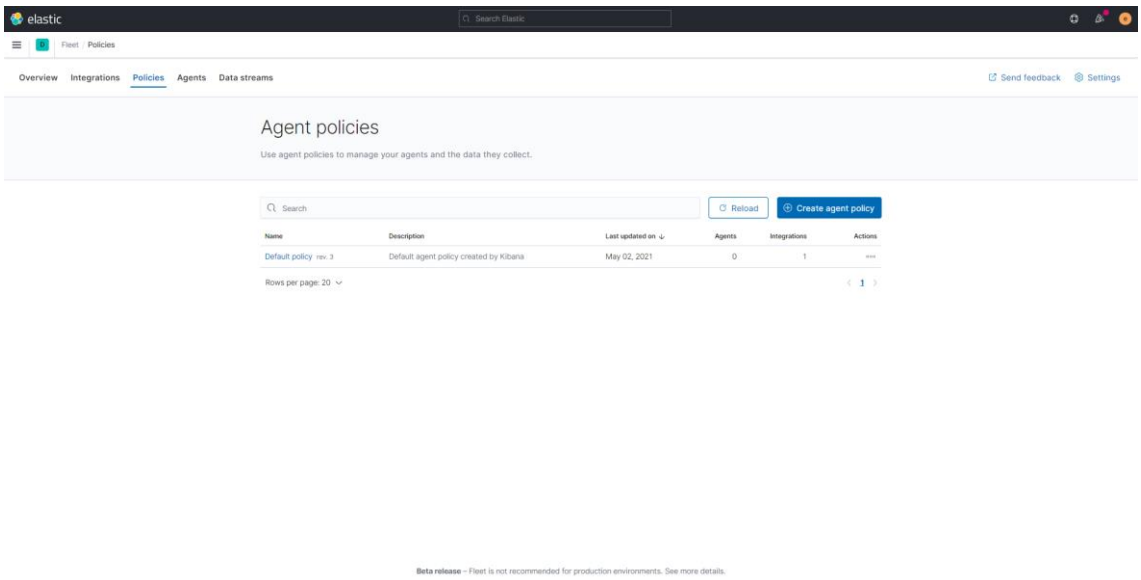

Figura 24. Fleet Policies.

Cada política tiene unos parámetros de configuración relativos a la recolección de datos del sistema anfitrión y a la monitorización de Elastic Agent y una serie de integraciones asociadas. Cada integración tiene su configuración específica y es posible que en algunos casos una política tenga asociada la misma integración más de una vez con configuraciones diferentes.

El tercer y último concepto relevante son los agentes, que se gestionan desde la pestaña de Agents. Desde esta pestaña se pueden obtener los comandos necesarios para añadir nuevos agentes así como visualizar una lista a modo resumen de todos los agentes desplegados.

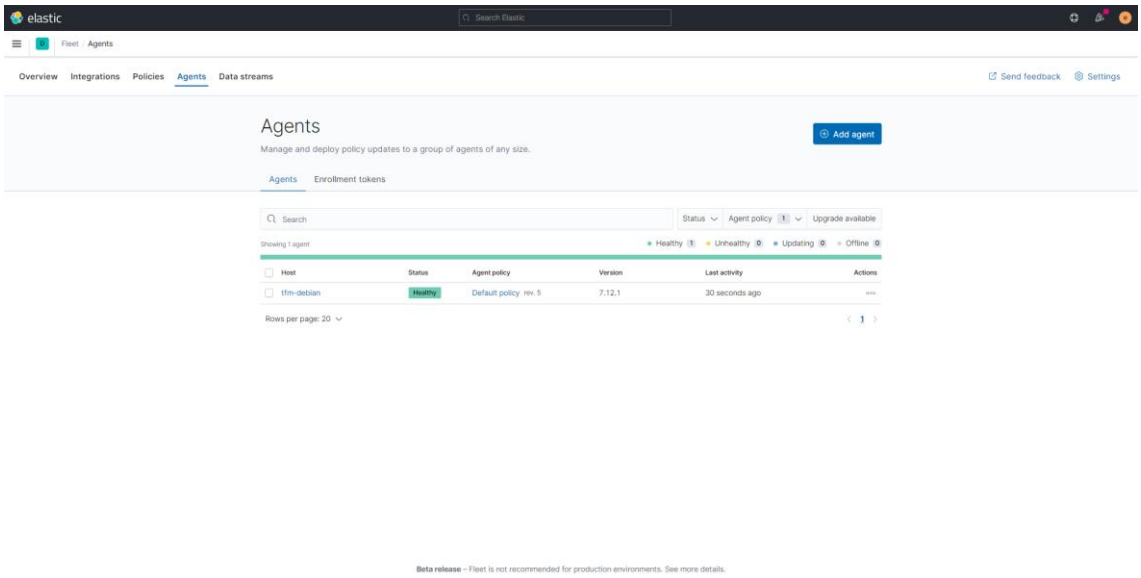

Figura 25. Fleet Agents.

Cada agente tiene asociada una política. Todas las modificaciones que se hagan sobre la política se trasladarán lo antes posible a los agentes asociados. Si los agentes están conectados esto será más o menos rápido y si no lo están, se actualizarán cuando recuperen la conectividad.

4.2 Endpoint Security

Como ya se ha mencionado anteriormente, Endpoint Security se distribuye como una integración de Fleet y por tanto su despliegue se realiza mediante Elastic Agent. La configuración concreta que se aplicará a cada endpoint dependerá por tanto de la configuración de la integración Endpoint Security de la política a la que esté asociado el agente del endpoint.

Parámetros

Al configurar Endpoint Security es posible seleccionar los tipos de eventos que se quieren recoger para cada uno de los sistemas operativos soportados, que son Windows, Mac y Linux.

Windows	**Mac**	**Linux**
DLL and Driver Load DNS File Network Process Registry Security	File Process Network	File Process Network

Protecciones

Al configurar Endpoint Security también es posible configurar dos tipos de protecciones. La primera es la protección contra malware, disponible en sistemas operativos Windows y Mac.

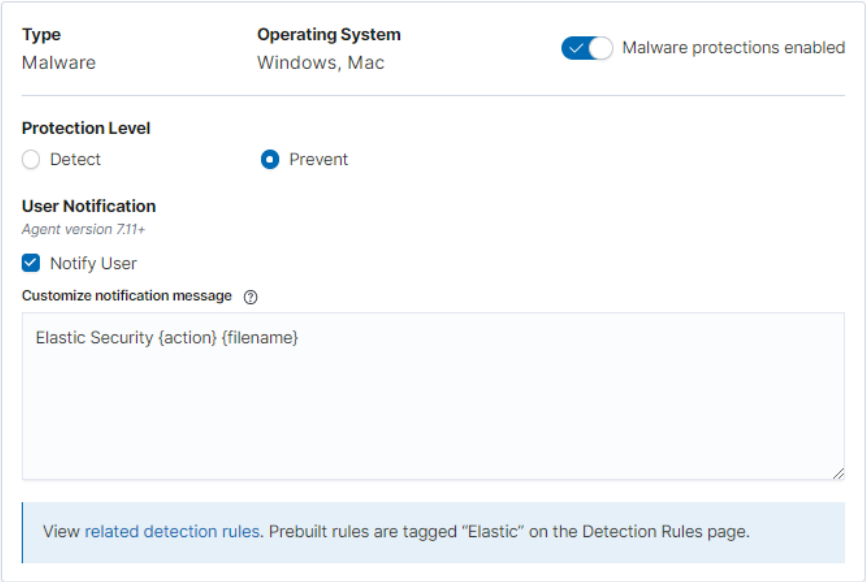

Figura 26. Protección contra Malware.

Se puede activar o desactivar. En caso de estar activada, se puede configurar también si se desea únicamente detectar el malware o se desea también prevenir su ejecución. En este último caso se puede notificar al usuario.

El segundo tipo de protección es contra ransomware y solo está disponible en equipos con sistema operativo Windows. Las opciones son las mismas que para la protección contra malware.

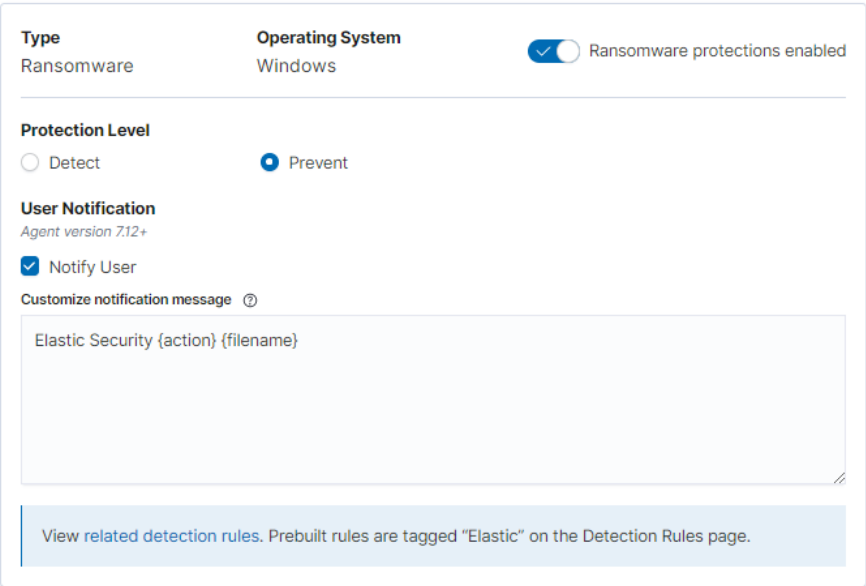

Figura 27. Protección contra Ransomware.

Además, es posible registrar Endpoint Security como antivirus en equipos con sistema operativo Microsoft Windows (salvo en ediciones Microsoft Windows Server). Esta opción hace que se desactive Windows Defender en estos equipos. Por defecto viene deshabilitada.

En cuanto a la evaluación, se ha hecho instalando Elastic Agent en dos máquinas virtuales, una con sistema operativo Microsoft Windows 10 y otra con sistema operativo Linux Debian 10.8 y con una política con dos integraciones con la configuración por defecto, System y Endpoint Security.

En cuanto a la máquina virtual con sistema operativo Debian Linux 10.8 se observa que se recogen eventos de los tres tipos de eventos soportados: File, Process y Network. Por ejemplo, si se ejecuta el siguiente comando:

```
romano@tfm-debian:~$ touch /var/tmp/test
```

se recoge el siguiente evento de tipo File:

```
{
  "_index" : ".ds-logs-endpoint.events.file-default-2021.05.02-000001",
  . . .
  "_source" : {
    . . .
    "file" : {
      "path" : "/var/tmp/test",
      . . .
    },
    . . .
    "event" : {
      . . .
      "category" : [
        "file"
      ],
      "type" : [
        "creation"
      ],
      . . .
    },
    "user" : {
      . . .
      "name" : "romano",
      "id" : 1000,
      . . .
    },
    . . .
  }
}
```

Y si se ejecuta el siguiente comando:

```
romano@tfm-debian:~$ curl https://www.google.es
```

se recoge el siguiente evento de tipo Process:

```
{
  "_index" : ".ds-logs-endpoint.events.process-default-2021.05.02-000001",
  . . .
  "_source" : {
    . . .
    "process" : {
      . . .
      "exit_code" : 0,
      "name" : "curl",
```

```
      "pid" : 2177,
      "command_line" : "curl https://www.google.es",
      "executable" : "/usr/bin/curl",
      . . .
    },
    . . .
    "event" : {
      . . .
      "category" : [
        "process"
      ],
      "type" : [
        "end"
      ],
      . . .
    }
    . . .
  }
}
```


Y el siguiente evento de tipo Network:


```
{
  "_index" : ".ds-logs-endpoint.events.network-default-2021.05.02-000001",
  . . .
  "_source" : {
    . . .
    "destination" : {
      "geo" : {
        "continent_name" : "North America",
        "country_iso_code" : "US",
        "country_name" : "United States",
        "location" : {
          "lon" : -97.822,
          "lat" : 37.751
        }
      },
      "address" : "142.250.184.163",
      "port" : 443,
      "ip" : "142.250.184.163"
    },
    . . .
    "network" : {
      "transport" : "tcp",
      "type" : "ipv4"
    },
    . . .
    "event" : {
      . . .
      "category" : [
        "network"
      ],
      "type" : [
        "end"
      ],
      . . .
    },
    . . .
  }
}
```


Al tener disponible en Elasticsearch está información es posible activar reglas de detección para detectar situaciones potencialmente peligrosas así como ayudar en la investigación de posibles incidentes.

Por ejemplo, se puede activar la regla para detectar conexiones FTP de cara a prevenir posibles exfiltraciones de información y luego hacer una conexión a un servidor FTP desde la máquina virtual `tfm-debian`.

Se comprueba que se generan alertas en el módulo de Detección:

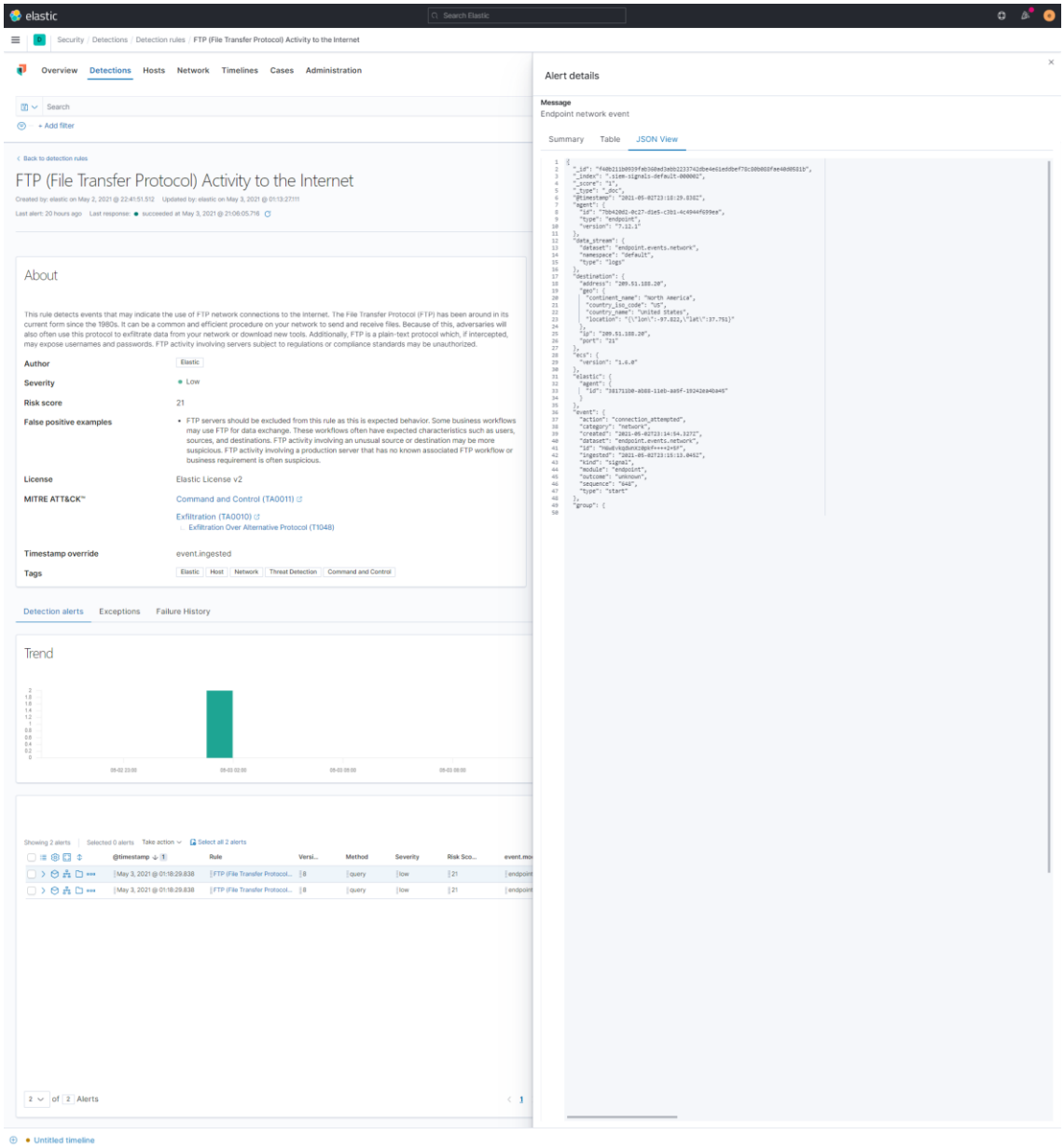

Figura 28. Alertas por conexiones FTP.

En cuanto a la máquina virtual con sistema operativo Microsoft Windows 10 se observa que se recogen eventos de todos los tipos soportados. Además, se prueba también la protección contra malware intentando ejecutar Mimikatz [39], observando que efectivamente Endpoint Security evita su ejecución.

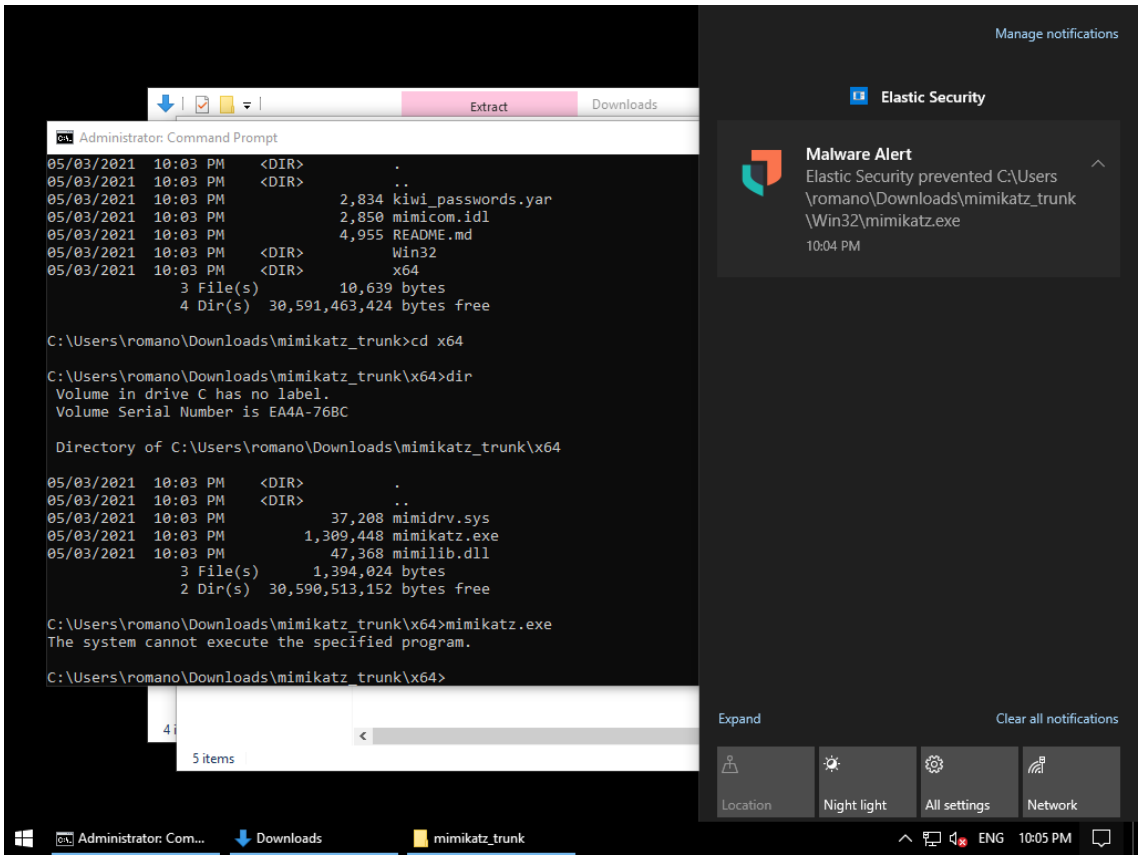

Figura 29. Protección contra Malware.

Y además, también se comprueba que la regla de Endpoint Security del módulo de Detección genera las alertas pertinentes:

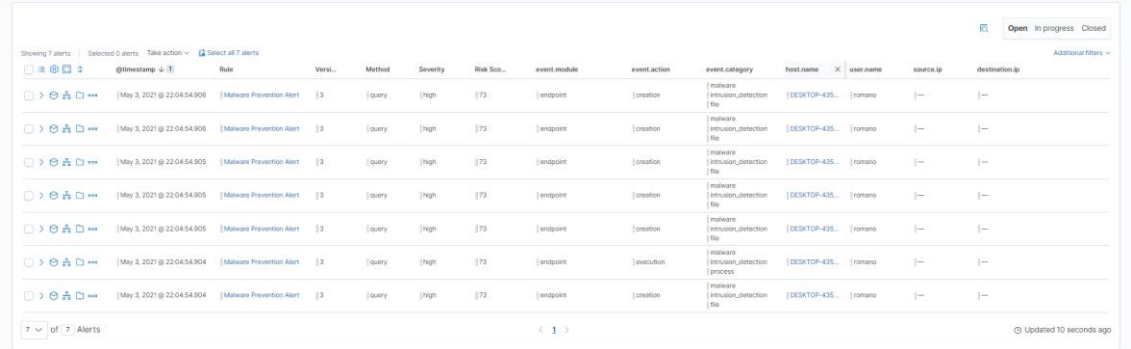

Figura 30. Alertas de Endpoint Security.

5. Despliegue

En los dos capítulos anteriores se han analizado en detalle y empleando un entorno de evaluación relativamente sencillo y acotado, las diferentes capacidades que Elastic Security puede ofrecer a una organización, como SIEM y como solución de protección de endpoint.

Pero ¿qué debe tener en cuenta una organización para desplegar estas capacidades en un entorno real? A lo largo de este capítulo se analizarán precisamente estos aspectos, intentando establecer unas pautas que puedan servir como guía a la hora de planificar un despliegue.

Elastic Stack

¿Existe en la organización una instalación de Elastic Stack que se pueda usar para Elastic Security? Si no existe una instalación de Elastic Stack que se pueda utilizar entonces lo primero que habrá que hacer es decidir qué tipo de instalación se quiere:

- Realizar una instalación *standalone* de Elastic Stack.
- Realizar una instalación utilizando Elastic Cloud on Kubernetes.
- Utilizar Elastic Cloud, la solución gestionada por Elastic.

En este Trabajo se ha optado por la segunda opción para el entorno de evaluación en donde se han probado las capacidades de Elastic Security. Pero luego cada organización puede tener ciertos requisitos que hagan más recomendable una determinada opción sobre otra:

- Si una organización tiene hardware disponible y no tiene experiencia con Kubernetes o con servicios en nubes públicas, la opción que parece más recomendable sería la primera.
- Si ya tiene un clúster de Kubernetes operativo y con capacidad disponible y el coste es un factor importante, seguramente la opción más recomendable sería la segunda.
- Si por el contrario la organización prefiere una solución gestionada por no disponer de personas que puedan operar adecuadamente la instalación, entonces la tercera parece la más lógica.

En cualquier caso, es importante que la opción elegida permite cumplir con los requisitos de rendimiento y de resiliencia. Además, hay que tener presente que la decisión no tiene que ser permanente, que es posible empezar con una determinada opción y migrar más adelante si parece más adecuado.

Por ejemplo, se puede empezar con la primera opción y migrar posteriormente a la segunda si la organización comienza a utilizar Kubernetes como solución de orquestación de contenedores.

Otro aspecto que se debe tener en cuenta es el conocimiento que la organización tenga de Elastic Stack. Elastic Security se apoya en muchas funcionalidades de Elastic Stack y por tanto, mantener adecuadamente un despliegue de Elastic Security requiere ciertos conocimientos de Elastic Stack.

Por ejemplo, es importante entender bien cómo funciona ILM (*Index Lifecycle Management*), el conjunto de mecanismos que ofrece Elastic Stack para mantener los índices de datos y definir políticas de ILM alineadas con los requisitos de rendimiento, resiliencia y retención de la organización.

La mayoría de fuentes de datos que se utilizarán en un despliegue de este tipo seguramente se modelen como *data streams* de Elasticsearch, una abstracción orientada a gestionar de forma eficiente series de datos temporales donde solamente se añade información.

Cada *data stream* está respaldado por un conjunto de índices físicos ocultos. Estos índices se pueden ir generando de forma manual mediante llamadas al API de *rollover* o se pueden generar de forma automática mediante políticas de ILM. Se puede hacer *rollover* por tiempo transcurrido desde la creación del índice, por número de documentos o por tamaño del índice.

Mediante políticas ILM también se pueden configurar estrategias *hot-warm-cold* e intentar así optimizar costes de infraestructura. Las series de datos temporales suelen ser muy buenas candidatas para este tipo de estrategias ya que la inserción se hace en un único índice (*hot*), la mayoría de las búsquedas se hacen sobre los últimos índices (*hot*) y el resto de búsquedas no suelen requerir tiempos de respuesta tan ajustados (*warm* o incluso *cold*).

Solamente hay que tener en cuenta que esta estrategia, sobre todo si se usa *cold data tier*, debe estar alineada con las restricciones que impone Elastic Security. Elastic Security no soporta el uso de *cold data tier* ni para índices internos (como los que se usan para almacenar alertas o listas de valores) ni para índices que se usen como fuentes de indicadores.

También es importante conocer bien ECS (*Elastic Common Schema*), que es una especificación abierta en la que se define qué campos se deben emplear para almacenar la información en los eventos y qué naturaleza deben tener estos campos y cuyo objetivo principal es permitir analizar, visualizar y correlar mejor la información almacenada en los eventos.

ECS facilita también unas directrices para los casos en los que haga falta adaptar un tipo de fuente de información a la especificación. Es importante que todo documento tenga un campo `@timestamp`, un campo `ecs.version` con la versión de la especificación utilizada y que se intenten reutilizar tantos campos previamente definidos como sea posible.

Por último, es importante también conocer las capacidades de gestión de privilegios, espacios y roles que ofrece Kibana y cómo se pueden utilizar en el caso en el que sea necesario controlar y segmentar el acceso a la información de Elastic Security.

Fuentes de Información

Otro punto importante es identificar qué información es la que se quiere consolidar en Elastic Stack. Es lo que se denominan fuentes de información, que son instancias específicas de ciertos tipos de información. Un tipo de información puede ser un log de acceso estándar de un servidor NGINX y una fuente de información puede ser el log concreto de un *virtual host* de uno de los servidores donde corre dicho NGINX.

Por cada fuente de información identificada hay que determinar cual es la mejor forma de llevar la información a Elastic Stack de forma normalizada (ECS). Las integraciones que incorpora Fleet parecen a priori la forma más sencilla de hacer esto pero también hay que tener en cuenta que solamente están soportadas ciertas soluciones y bajo ciertas condiciones.

Es importante analizar los requisitos de cada integración ya que pueden condicionar el número de agentes requeridos para integrar todas las fuentes de información de un mismo tipo. Por ejemplo, si se quieren integrar los logs de dos cuentas de AWS puede ser necesario desplegar dos agentes, cada uno con una configuración diferente de la integración AWS.

Fleet permite también la ingesta de logs genéricos mediante la integración Custom Logs. Al contrario que otras integraciones de Fleet, esta integración sí que permite configurar una pipeline de ingestión para adaptar en la medida de lo posible esos logs genéricos a formato ECS [41].

Para otras soluciones todavía no soportadas por Fleet será necesario emplear Beats directamente o utilizar otras alternativas como Fluentd [40] o Logstash. Como ya se ha mencionado será conveniente que esta alternativa sea capaz de ingestar la información en formato ECS. Por ejemplo, si se quieren usar indicadores de fuentes como abuse.ch o AlienVault OTX se puede usar el módulo de Threat Intel de Filebeat.

Es importante también que la incorporación de fuentes de información se haga de forma gradual, prestando especial atención al dimensionamiento del clúster de Elastic Stack. Por lo general, más fuentes de información requieren más capacidad de procesamiento y requieren más almacenamiento en todas sus variantes *hot*, *warm* y *cold*.

En resumen:

1. Elaborar una lista de fuentes de información a integrar.
2. Analizar la mejor forma de integrar cada tipo de fuente de información.
3. Añadir fuente de información.
4. Monitorizar el clúster y asegurarse de que dispone de recursos.
5. Iterar sobre los puntos 3 y 4 hasta integrar todas las fuentes.

Endpoint Security

Lo siguiente que se debe tener en cuenta es si se quieren utilizar las capacidades de detección y protección de endpoint que ofrece Elastic Security y, sobre todo, cómo se quieren utilizar, lo cual dependerá en gran medida de otras iniciativas que pueda haber en la organización.

A priori, la recolección de eventos no parece que pueda entrar en conflicto con otras iniciativas, pero siempre va a suponer un coste asociado al procesado y al almacenamiento de estos eventos. Por tanto, la recomendación es activar la recolección de eventos cuando se de al menos una de estas circunstancias: se quieren usar las capacidades de SIEM de Elastic Security o estos eventos no se están almacenando en ninguna otra plataforma.

Si se va a activar la recolección de eventos lo que sí es importante es hacerlo de forma gradual, ya que cada agente va a demandar recursos tanto para la ingestión como para la consulta y el almacenamiento de los eventos. Se puede desplegar primero un número controlado de agentes y observar durante un tiempo el comportamiento del clúster. Si es adecuado, se puede desplegar un nuevo conjunto de agentes. Pero si se observan problemas de rendimiento, entonces será necesario ampliar el clúster antes de seguir con el despliegue.

En cuando a las protecciones, lo primero que se debe tener presente es que éstas son distintas según el sistema operativo. Por ejemplo, en Windows está disponible tanto la protección contra *malware* como contra *ransomware* pero en Mac solo está disponible la protección contra *malware* y en Linux no está disponible ninguna de las dos.

Una ventaja de la aproximación de Elastic, también presente en otros muchos productos, es que permite bastante granularidad a la hora de configurar cómo se quieren usar las protecciones. Por ejemplo, se pueden definir varias políticas en Fleet, cada una con una configuración de Endpoint Security distinta:

- Con las protecciones contra *malware* y *ransomware* desactivadas.
- Con las protecciones contra *malware* y *ransomware* en modo Detect.
- Con las protecciones contra *malware* y *ransomware* en modo Prevent.

y luego asociar a cada agente la política con la configuración que más interese en cada momento. En cualquier caso, es importante probar primero estas políticas en entornos controlados, antes de desplegarlas de forma masiva.

Por ejemplo, si se trata de una política que se quiere desplegar en servidores, se puede desplegar primero en entornos de prueba o de preproducción. Si se trata de una política para equipos de usuario, se puede elegir primero un número reducido de equipos donde desplegarla primero.

De esta forma, es más sencillo controlar que las protecciones de endpoint que se desplieguen no impacten negativamente en el funcionamiento de las aplicaciones de la organización o, que si impactan, se pueda detectar a tiempo y se puedan añadir como aplicaciones confiables de Elastic Security.

SIEM

Por último quedaría decidir qué uso es el que la organización quiere hacer de las capacidades de SIEM de Elastic Security. Y lo primero a decidir es sí se va a emplear el módulo de detecciones y si es así, qué reglas de detección son las que se van a utilizar.

Como ya se ha visto en el tercer capítulo, junto con Elastic Security se distribuye un conjunto de reglas de detección bastante variado. Por ejemplo, en la versión 7.12 son más de 540 reglas las que se pueden instalar:

- Si se está usando Endpoint Security hay una regla que nos permite crear una alerta de detección siempre que se produzca una alerta de Endpoint Security en alguno de los *endpoints* protegidos.

- El resto de las reglas están etiquetadas según el ámbito en el que aplican o el tipo de situación que pretenden detectar. Por ejemplo, las reglas relacionadas con IAM de AWS tienen etiquetas como AWS o Cloud.

Lógicamente estas reglas se pueden complementar con reglas desarrolladas por la propia organización para detectar situaciones que para la organización sean relevantes o potencialmente peligrosas pero todavía no estén en el conjunto que se distribuye con Elastic Security.

En cualquier caso, es importante que antes de activar cualquier regla se haya entendido bien qué situación es la que pretende detectar, por qué esta situación es potencialmente peligrosa para la organización y sobre todo, qué pasos se deben seguir si se producen alertas de detección asociadas a esta regla para determinar si se trata de un incidente de seguridad o no.

Además, se debe contar con que la activación de cualquier regla de detección puede generar al principio un número considerable de alertas, muchas de las cuales es probable que tras el análisis inicial acaben siendo realmente falsos positivos y por tanto, convenga ir añadiendo excepciones para ir refinando así la regla de detección.

El otro tema relevante para el despliegue de las capacidades de SIEM es determinar cómo va a interactuar la organización con las alertas de detección, que estará condicionado en gran medida con la licencia de Elastic Stack de la que disponga.

El escenario más sencillo es disponer de una licencia que permita al menos ejecutar acciones cuando se creen nuevas alertas de detección, de tal forma que se pueda notificar a un buzón de correo electrónico que existen nuevas alertas y que es necesario analizarlas.

A partir de ese punto, las capacidades de Elastic Security pueden ser suficientes, ya que las alertas pueden marcarse como In Progress o Closed y se pueden asociar a Casos, que permiten llevar una bitácora mucho más completa de las acciones realizadas.

Si la organización no dispone de al menos una licencia *gold* todavía tiene dos opciones para detectar nuevas alertas. La más obvia pero muy poco eficiente es entrar periódicamente en la consola de Elastic Security y comprobar si hay nuevas alertas de detección desde la última vez que se revisaron.

Por lo tanto, lo que en este caso puede tener más sentido es lo que ya se comentaba en el tercer capítulo, que la organización desarrolle un proceso que mediante API vaya obteniendo las alertas creadas desde la última consulta y envíe un correo electrónico a modo de notificación.

6. Conclusiones

Este Trabajo partía de la premisa de que Elastic Security podía ser un buen candidato para que una organización desplegase una solución de seguridad que le permitiera de forma más o menos sencilla dotarse de capacidades de detección, prevención y análisis de incidentes de seguridad.

Para verificar esta premisa se proponía desplegar un entorno de evaluación donde poder probar en detalle las diferentes funcionalidades que ofrecía la solución para determinar si podían ser suficientes y cómo una organización podía desplegarlas en su perímetro.

En ese sentido, se puede concluir que la premisa era correcta y que Elastic Security es, efectivamente, una buena alternativa para que una organización implemente una solución de detección, prevención y análisis de incidentes de seguridad:

- Ofrece capacidades de detección y protección para endpoint que complementan las soluciones tradicionales de antivirus. Recogen telemetría de los endpoints y además, para Windows y Mac ofrecen protección basada en comportamiento.

- Las capacidades de detección de posibles incidentes son también bastante completas, desde consultas simples sobre índices de Elastic a reglas más complejas basadas en EQL o en *machine learning*.

- Estos posibles incidentes se pueden analizar en detalle en la propia solución, con visualizaciones de ciertos tipos de eventos bastante explicativas o la posibilidad de creación de plantillas para facilitar el trabajo.

- Y además, si se tiene la licencia adecuada es posible mantener sincronizado este análisis de los posibles incidentes con las herramientas de *ticketing* de la organización.

- Y todo eso con un producto que gracias a Fleet ha simplificado mucho tanto el despliegue de las capacidades de endpoint como la recolección y la normalización de logs y de métricas de soluciones de terceros.

En cuanto a los objetivos que se establecían al inicio del Trabajo, también se puede afirmar que se han cumplido. El primer objetivo era que el entorno de evaluación que se emplease se pudiera trasladar de forma sencilla a un entorno tipo cloud que fuera posible escalar en función de las necesidades de cada organización.

El soporte de Kubernetes por parte de Elastic mediante su operador ECK ha facilitado mucho el cumplimiento de este objetivo. Kubernetes es una solución de orquestación de contenedores soportada en la mayoría de nubes públicas y privadas y que además es fácilmente escalable para cumplir con los requisitos de cada organización.

Lo mismo que se ha descrito para el entorno de evaluación montado utilizando un equipo doméstico y minikube se puede ejecutar sobre cualquiera de los servicios gestionados de Kubernetes que ofrecen los principales proveedores de nubes públicas, como Azure Kubernetes Service (AKS), Amazon Elastic Kubernetes Service (EKS) o Google Kubernetes Engine (GKE).

Este tipo de servicios simplifica la gestión del plano de control de Kubernetes y junto con ECK permite a la organización centrarse en la selección de los recursos que quiere dedicar a la instalación de Elastic, en función de los requisitos y del presupuesto disponible, que condicionarán el tipo de nodos a emplear para los diferentes roles que requiere Elastic y el tipo de almacenamiento para datos hot, warm y cold.

Por ejemplo, los datos de la última semana se pueden tener en nodos de consulta con almacenamiento de altas prestaciones, los datos de los últimos tres meses en nodos con menos prestaciones y con almacenamiento más barato y los datos históricos que se deban almacenar por cuestiones regulatorias o normativas, en forma de *snapshots* en soluciones de bajo coste como S3.

El segundo objetivo era probar las capacidades de detección y de análisis de la solución. Se han creado reglas de todos los tipos soportados por la solución, incluidos los que permiten usar feeds de terceros y se ha verificado su funcionamiento. Se han probado también las líneas de tiempo y gestión de casos.

El tercer objetivo era analizar las capacidades de integración de la solución. Se ha podido verificar que con una licencia adecuada se tiene acceso a los conectores externos de Kibana que permiten, por ejemplo, enviar correos electrónicos o invocar URLs externas a modo de *webhook*.

Se ha podido verificar también que aunque no se disponga de esa licencia, con algo de desarrollo una organización puede emular parcialmente esas capacidades, mediante consultas periódicas a las APIs que ofrece Elastic Security y programando ella las acciones mencionadas.

Por último, se ha podido verificar también el cuarto objetivo, probar las capacidades de detección y protección de endpoint. Se ha comprobado que se recibe telemetría de los diferentes sistemas operativos y que en endpoints con sistema operativo Windows se detecta malware como Mimikatz.

En cuanto a la planificación del Trabajo, si ha sido necesario realizar ciertos cambios durante el desarrollo del mismo. Por ejemplo, en el plan de trabajo inicial se proponía evaluar primero las capacidades de endpoint y el tiempo asignado a la evaluación de las capacidades de SIEM no era el más adecuado.

En cuanto a las líneas de trabajo futuro, son varias. La primera, probar algunas funcionalidades que no se han podido evaluar en este Trabajo, como la protección de endpoint para Mac o que se han evaluado solo parcialmente, como las reglas de detección basadas en EQL o en trabajos ML.

En el caso de EQL, es un lenguaje que ofrece capacidades avanzadas de análisis de eventos que permiten detectar amenazas a partir de correlaciones com-

plejas. Pero no es sencillo, por lo que pare crear reglas efectivas y sobre todo, con un rendimiento aceptable, es necesario conocerlo a fondo.

Y el caso de los trabajos Machine Learning para detección de anomalías pasa algo parecido. La parte relacionada con Elastic Security es bastante simple pero la definición y configuración de los trabajos no lo es. El potencial es bastante grande pero también requiere un conocimiento profundo para sacarle partido.

Tampoco se han probado los conectores externos con sistemas como Atlassian Jira o Service Now. Si el despliegue de la solución se hace en una organización que use alguna de estas herramientas de ticketing si puede ser interesante analizarlos en más en detalle.

Además, con cada versión de Elastic, que se publica cada pocos meses, se incluyen nuevas funcionalidades relacionadas con Elastic Security o con Fleet. Por tanto, también es necesario estar atento a estas nuevas versiones para analizar lo que pueden aportar a la organización.

7. Glosario

API. Application Programming Interface. Capa de abstracción que ofrece una aplicación para que terceros puedan utilizar un subconjunto de sus funcionalidades.

DSL. Domain Specific Language. Lenguaje de programación o especificación dedicado a resolver un problema en particular.

EDR. Endpoint Detection and Response. Conjunto de tecnologías que monitorizan de forma continua los equipos con el objetivo de detectar actividades sospechosas.

EQL. Event Query Language. Un lenguaje especializado en consultas sobre series de datos temporales como logs, métricas y trazas. Originalmente diseñado por Endgame e incorporado en Elastic Stack tras adquirir la compañía.

GPL. GNU General Public License. Conjunto de licencias de software libre que permiten garantizar que los usuarios tengan libertad para ejecutar, modificar y compartir y software.

REST. Representational State Transfer. Paradigma para construir interfaces entre sistemas sin estado y basados fundamentalmente en el uso de recursos HTTP e hipermedios.

SIEM. Security Information and Event Management. Sistemas que centralizan el almacenamiento y la interpretación de los datos relevantes desde un punto de visto de seguridad.

SOAR. Security Orchestration, Automation and Response. Conjunto de tecnologías orientadas a que las organizaciones puedan automatizar la gestión de amenazas, incidentes y vulnerabilidades.

8. Bibliografía

[1] Elastic Stack
https://www.elastic.co/elk/stack
Consultado el 26 de marzo de 2021.

[2] Beats
https://www.elastic.co/beats/
Consultado el 26 de marzo de 2021.

[3] Logstash
https://www.elastic.co/logstash
Consultado el 26 de marzo de 2021.

[4] Elasticsearch
https://www.elastic.co/es/elasticsearch/
Consultado el 26 de marzo de 2021.

[5] Kibana
https://www.elastic.co/es/kibana
Consultado el 26 de marzo de 2021.

[6] Elastic Security
https://www.elastic.co/guide/en/security/current/index.html
Consultado el 26 de marzo de 2021.

[7] Elastic Blog. Elasticsearch 7.12 released.
https://www.elastic.co/blog/whats-new-elasticsearch-7-12-0-put-a-search-box-on-s3
Consultado el 26 de marzo de 2021.

[8] Elastic Cloud on Kubernetes
https://www.elastic.co/guide/en/cloud-on-k8s/current/index.html
Consultado el 28 de marzo de 2021.

[9] minikube
https://minikube.sigs.k8s.io/
Consultado el 28 de marzo de 2021.

[10] Elastic Endpoint Security
https://www.elastic.co/endpoint-security/
Consultado el 28 de marzo de 2021.

[11] GanttProject
https://www.ganttproject.biz/
Consultado el 28 de marzo de 2021.

[12] Kubernetes
https://kubernetes.io
Consultado el 28 de marzo de 2021.

[13] Elastic Cloud
https://www.elastic.co/cloud
Consultado el 28 de marzo de 2021.

[14] RedHat OpenShifht
https://www.redhat.com/en/technologies/cloud-computing/openshift
Consultado el 28 de marzo de 2021.

[15] Oracle VirtualBox
https://www.virtualbox.org/
Consultado el 28 de marzo de 2021.

[16] Docker Engine
https://docs.docker.com/engine/
Consultado el 28 de marzo de 2021.

[17] kubectl
https://kubernetes.io/docs/reference/kubectl/kubectl/
Consultado el 28 de marzo de 2021.

[18] Let's Encrypt
https://letsencrypt.org/
Consultado el 28 de marzo de 2021.

[19] certbot
https://certbot.eff.org/
Consultado el 28 de marzo de 2021.

[20] Electronic Frontier Foundation
https://www.eff.org/
Consultado el 28 de marzo de 2021.

[21] Elastic Detection Rules
https://github.com/elastic/detection-rules
Consultado el 30 de marzo de 2021.

[22] Kibana Query Language
https://www.elastic.co/guide/en/kibana/current/kuery-query.html
Consultado el 30 de marzo de 2021.

[23] Lucene Query Language
https://www.elastic.co/guide/en/kibana/current/lucene-query.html
Consultado el 30 de marzo de 2021.

[24] Event Query Language
https://www.elastic.co/guide/en/elasticsearch/reference/7.12/eql.html
Consultado el 30 de marzo de 2021.

[25] ATT&CK
https://attack.mitre.org
Consultado el 19 de abril de 2021.

[26] MITRE
https://www.mitre.org
Consultado el 19 de abril de 2021.

[27] Machine Learning in the Elastic Stack
https://www.elastic.co/guide/en/machine-learning/current/index.html
Consultado el 23 de abril de 2021.

[28] Filebeat Threat Intel module
https://www.elastic.co/guide/en/beats/filebeat/7.12/filebeat-module-threatintel.html
Consultado el 20 de abril de 2021.

[29] abuse.ch
https://abuse.ch
Consultado el 20 de abril de 2021.

[30] AlienVault OTX
https://otx.alienvault.com/
Consultado el 20 de abril de 2021.

[31] Elastic Common Schema
https://www.elastic.co/guide/en/ecs/current/index.html
Consultado el 25 de abril de 2021.

[32] ServiceNow
https://www.servicenow.com
Consultado el 25 de abril de 2021.

[33] Jira Software
https://www.atlassian.com/software/jira
Consultado el 25 de abril de 2021.

[34] Jira Service Management
https://www.atlassian.com/software/jira/service-management
Consultado el 25 de abril de 2021.

[35] IBM Resilient
https://www.ibm.com/products/soar-platform
Consultado el 25 de abril de 2021.

[36] Elasticsearch Query DSL
https://www.elastic.co/guide/en/elasticsearch/reference/7.12/query-dsl.html
Consultado el 2 de mayo de 2021.

[37] Kibana Saved Objects API
https://www.elastic.co/guide/en/kibana/7.12/saved-objects-api-find.html
Consultado el 2 de mayo de 2021.

[38] Fleet User Guide
https://www.elastic.co/guide/en/fleet/7.12/index.html
Consultado el 2 de mayo de 2021.

[39] Mimikatz
https://github.com/gentilkiwi/mimikatz/wiki
Consultado el 3 de mayo de 2021.

[40] Fluentd
https://www.fluentd.org/
Consultado el 10 de mayo de 2021.

[41] Elasticsearch Ingest Pipelines
https://www.elastic.co/guide/en/elasticsearch/reference/7.12/ingest.html
Consultado el 11 de mayo de 2021.

9. Anexos

9.1 Entorno de Demostración

Para ilustrar de forma práctica lo que se expone en esta memoria se ha optado por implementar un caso de uso a partir del entorno de evaluación descrito en el segundo capítulo.

El caso de uso es sencillo y consiste en detectar si se ha accedido a URLs clasificadas como malware por abuse.ch desde algún PC de usuario de la organización, teniendo en cuenta que el acceso a Internet se realiza a través de un servidor proxy basado en Squid.

Si se revisan las consideraciones expuestas en el quinto capítulo y suponiendo que ya se dispone del entorno de Elastic Stack y de los conocimientos necesarios para operarlo, el primer paso sería identificar las fuentes de información que debemos integrar:

- La información de URLs de malware de abuse.ch
- Los logs de acceso del servidor proxy basado en Squid

Para integrar la información de URLs de malware la forma más sencilla parece ser usando el módulo de Threat Intel de Filebeat. Tras seguir los pasos descritos en la guía de instalación y configuración de Filebeat podemos comprobar que en Elasticsearch se están cargando eventos de tipo `indicator` con URLs de malware.

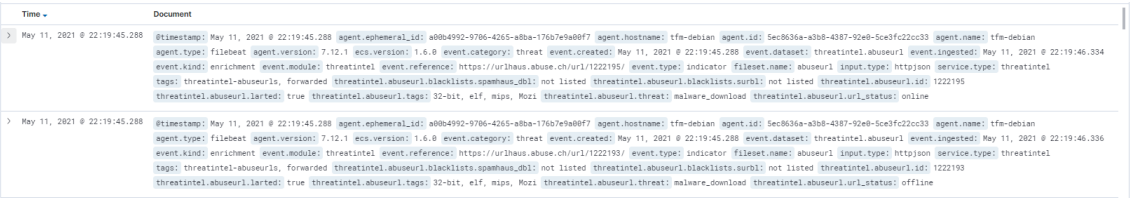

Figura 31. Eventos con URLs de malware de abuse.ch.

Para integrar los logs de acceso del servidor proxy basado en Squid se usará la integración de Fleet. Para ello se instalará la integración y se añadirá a la política por defecto, política que se asociará al agente que se instalará en la máquina virtual Linux Debian del entorno de evaluación, que es donde se está ejecutando el servidor proxy.

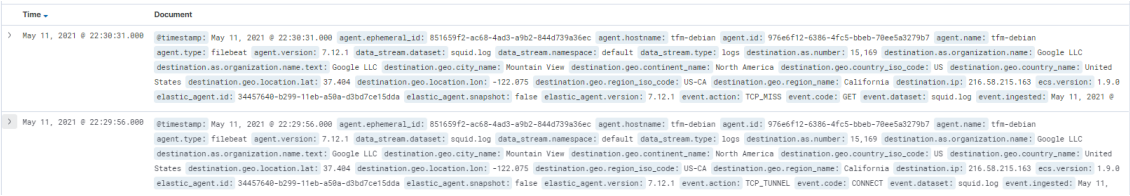

Figura 32. Eventos de acceso del servidor proxy.

A continuación se procederá a instalar en la máquina virtual Windows del entorno de evaluación un agente de Elastic con una política en donde se ha configurado Endpoint Security para recoger todos los tipos de eventos y para detener ataques tanto de *malware* como de *ransomware*.

El siguiente paso será crear una regla de detección de tipo Indicator Match para detectar accesos a URLs listadas como de malware por abuse.ch:

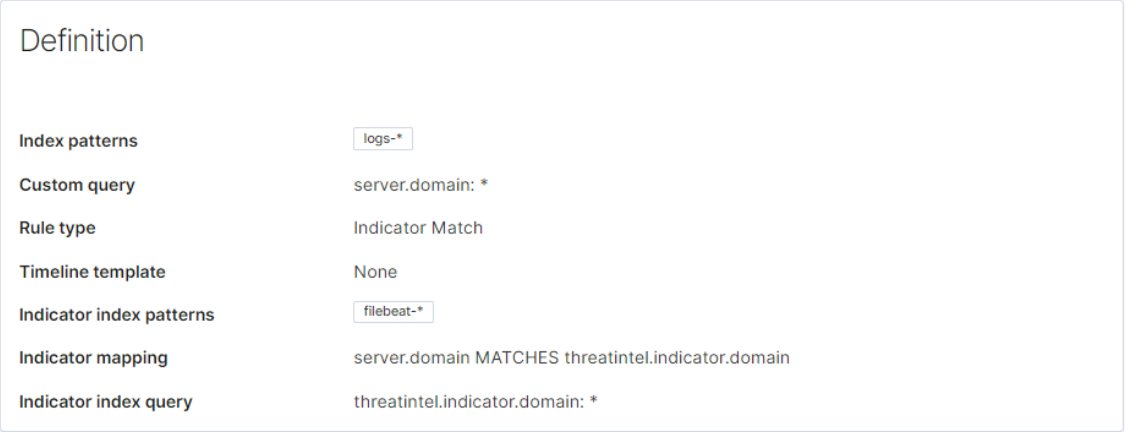

Figura 33. Regla para detectar accesos a URLs de malware.

Desde la máquina virtual Windows se hace un acceso con Microsoft Edge configurado para utilizar el servidor proxy basado en Squid a una de las URLs listadas como malware por abuse.ch. Se comprueba que la regla detecta el acceso y crea la correspondiente alerta:

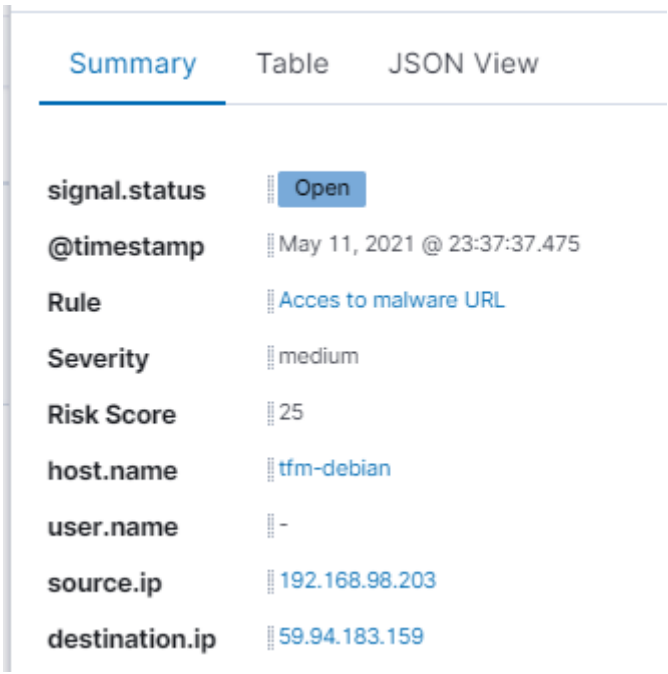

Figura 34. Alerta de acceso a URL de malware.