UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Desarrollo de SIEM para la detección de amenazas

Trabajo Fin de Máster

Máster Universitario en Ciberseguridad y Ciberinteligencia

AUTOR/A: Santos Ortega, Paula

Tutor/a: Esteve Domingo, Manuel

CURSO ACADÉMICO: 2022/2023

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

Escola Tècnica Superior d'Enginyeria Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Desarrollo de SIEM para la detección de amenazas

TRABAJO FIN DE MÁSTER

Máster Universitario en Ciberseguridad y Ciberinteligencia

Autora: Paula Santos Ortega

Tutor: Manuel Esteve Domingo

Curso 2022-2023

Resumen

El inminente crecimiento de la digitalización de las empresas y el uso de las tecnologías como actividad diaria dentro de ellas ha provocado que estas organizaciones se encuentren expuestas a ataques cibernéticos. Debido a esto, la ciberseguridad es primordial para detectar las amenazas a las que se enfrenta una empresa y para proteger los datos y la infraestructura de la misma. Este Trabajo de Fin de Máster tiene como objetivo desarrollar una plataforma SIEM que recopile toda la información obtenida de diferentes fuentes desplegadas en una empresa, para facilitar al equipo de Ciberseguridad la detección de amenazas así como la respuesta ante incidentes. Se investigará sobre qué fuentes se encuentran disponibles actualmente y cuáles ofrecen las características necesarias para integrarlos en el SIEM.

Palabras clave: Amenazas, fuentes de datos, correlación, SIEM

Resum

L'imminent creixement de la digitalització de les empreses i l'ús de les tecnologies com a activitat diària ha provocat que aquestes organitzacions estiguin exposades a atacs cibernètics. A causa d'això, la ciberseguretat és primordial per detectar les amenaces a què s'enfronta una empresa i per protegir les dades i la infraestructura. Aquest Treball de Fi de Màster té com a objectiu desenvolupar una plataforma SIEM que recopili tota la informació obtinguda de diferents fonts desplegades en una empresa, per facilitar a l'equip de Ciberseguretat la detecció d'amenaces així com la resposta davant d'incidents. S'investigarà sobre quines fonts es troben disponibles actualment i quines ofereixen les característiques necessàries per integrar-los al SIEM.

Paraules clau: Amenaces, fonts de dades, correlació, SIEM

Abstract

The imminent growth of the digitalization of companies and the use of technologies as a daily activity within them has caused these organizations to be exposed to cyber attacks. Because of this, cybersecurity is paramount to detect the threats that a company faces and to protect the data and infrastructure of the company. This Master Thesis aims to develop a SIEM platform that collects all the information obtained from different sources deployed in a company, to facilitate the Cybersecurity team the detection of threats as well as the response to incidents. Research will be done on which sources are currently available and which ones offer the necessary features to integrate them into the SIEM.

Key words: Threats, data sources, correlation, SIEM

Índice general

Índice de figuras

Índice de tablas

Agradecimientos

Dejé atrás mi querida Andalucía para venir aquí a València a cursar este máster en Ciberseguridad y Ciberinteligencia. Vine con mis maletas, mi portátil y mi gato Otoño. Quién me iba a decir que, dos años después, ya habría comenzado una vida aquí, con nuevos entornos, nuevas personas y nuevas energías.

Con este proyecto, finalizo una de las etapas más enriquecedoras de mi vida, mi etapa como estudiante. Y solo puedo dar las gracias.

A la UPV,
por brindarme la oportunidad
de ser una de las alumnas de este máster.

A València,
por todo lo que ha supuesto,
y lo que queda por ser.

A Mireia y a Sergio,
por vosotros.

A mi familia,
a mi madre,
a mi hermano,
por estar incluso en la distancia.

A mi pareja,
por acompañarme siempre de la mano,
por todo lo que somos.

A mi gato y a mi gata,
por y con todo el cariño.

A mí misma,
por estar en constante evolución
y en busca de nuevos retos,
y, especialmente,
por no tener fuerzas para rendirme.

CAPÍTULO 1
Introducción

La seguridad de los equipos informáticos y de la información de una organización se ha convertido en una parte esencial en las empresas. Cualquier compañía, independientemente de su tamaño, es susceptible de sufrir un ataque cibernético. En los últimos años, se ha producido un aumento significativo de los ciberataques a nivel mundial, estimándose en 2023 un incremento del 28 % en comparación con el año anterior [1].

Estos ataques, al igual que las tecnologías, infraestructuras y el almacenamiento de los datos, han ido evolucionado con los años. En los comienzos de la informática, los activos vulnerables eran los equipos y la documentación en papel, por tanto, la seguridad era física basada en personas que hacían de guardias para controlar el acceso a las instalaciones [2]. Hoy en día, además de proteger las instalaciones, también hay que tener en cuenta las redes que se utilizan, la información que se maneja y las diferentes aplicaciones y herramientas con las que se trabaja. Para proteger todo esto se utilizan mecanismos de protección y defensa más avanzados y complejos.

Dada la complejidad para gestionar la seguridad de una empresa, estas contratan un servicio SOC (*Security Operations Center*), que se trata de una central de seguridad informática que previene, monitorea y controla la seguridad en las redes y en Internet. Para poder realizar esto, centralizan todas las herramientas desplegadas en una plataforma SIEM (Security Information and Event Management), que facilita a los trabajadores del SOC la detección de comportamientos sospechosos y patrones no habituales en tiempo real desde una perspectiva global.

1.1 Motivación

Hay múltiples razones que me han llevado a la elección de este asunto para este proyecto de fin de máster. Se trata de un tema relevante y actual, que se encuentra en constante cambio y que ha afectado, y está afectando, con gran impacto a organizaciones importantes a nivel nacional e internacional. A diario emergen nuevas amenazas y ataques por parte de los ciberdelincuentes, y también se implementan nuevas herramientas y soluciones para poder defenderse.

Otro de los motivos es poder descubrir cómo funciona un SIEM desde la parte de arquitectura, cómo operan los detectores, cómo se integran, cómo se generan las reglas de detección.

1.2 Objetivos

El propósito principal de este proyecto es el desarrollo de una plataforma SIEM que facilite la detección de comportamientos anómalos y permita agilizar la respuesta ante incidentes.

Los objetivos a alcanzar son, por tanto, los siguientes:

1. Investigar qué herramientas existen actualmente, conocer su funcionamiento y escoger las que mejores prestaciones ofrezcan.
2. Implementar una plataforma SIEM que concentre todos los datos relevantes de seguridad.
3. Experimentar con la plataforma realizada para comprobar el correcto funcionamiento y corregir los posibles errores.

1.3 Estructura

Para finalizar esta introducción, se explicará los diferentes apartados que se abordarán en esta memoria:

- **Estudio y selección de herramientas:** se muestra las herramientas que hay disponibles para proteger y responder, la investigación de cada una de los productos probados, una comparativa entre ellas y la selección de cuáles se van a emplear para el proyecto.
- **Implementación:** se expone la instalación y configuración de los productos, así como su integración.
- **Pruebas:** se documentan las pruebas efectuadas para la comprobación del buen funcionamiento así como la corrección de errores.
- **Conclusiones y trabajos futuros:** se informa de la resolución y se concreta una valoración personal y los trabajos futuros.
- **Bibliografía:** se referencia toda la documentación.

CAPÍTULO 2
Estudio y selección de herramientas

2.1 Tecnologías de monitorización y detección

Para poder monitorizar la actividad de una empresa, es necesario el uso de herramientas que permitan obtener información acerca de qué está ocurriendo en los activos.

A continuación, se muestra un listado de algunos de los recursos que ayudan a detectar posibles ataques y pueden responder ante estos.

- **Firewall**: el cortafuegos permite filtrar el tráfico, tanto entrante como saliente, en una empresa y permite o bloquea un tráfico específico según una serie de políticas de seguridad definidas [3]. El firewall es la primera barrera de defensa, a nivel de red, de cualquier organización; evitan ataques de denegación de servicio (Denial of Service (DoS)), rechaza conexiones no autorizadas y protegen la información de los equipos.

  Palo Alto Networks [4], Fortinet [5] y Cisco [6] son tres grandes compañías que ofrecen firewalls potentes.

- **IDS *(Intrusion Detection System)***: estos sistemas monitorizan la actividad para detectar actividad no habitual o accesos no autorizados. Cuando se detecta actividad sospechosa o usual de ataque, generan alertas con la información recabada para que las personas administradoras tomen las medidas oportunas. Estos IDS pueden implementarse en diferentes entornos [7]:

  - **HIDS *(Host-Based IDS)***: se implementa en un equipo en el que observa los procesos en ejecución, inspecciona los registros del sistema y los *demonios*, entre otras tareas. Aunque se limita a un dispositivo, tiene una visibilidad profunda de lo que ocurre en sus componentes internos.

    OSSEC [8], Samhain [9] y Security Onion [10] son tres herramientas que actúan como HIDS.

  - **NIDS *(Network-Based IDS)***: se implementa en la red. Se encarga de la supervisión de una red, analizando y filtrando el tráfico. Detecta anomalías y comportamientos maliciosos, como acceso a sitios web maliciosos o ataques de *man-in-the-middle*.

    Algunos de los NIDS más reconocidos son Snort [11], Suricata [12] y Zeek (Bro) [13].

- **Antivirus**: son programas dedicados a detectar y bloquear acciones maliciosas en un equipo generadas por un malware. En caso de infección, elimina la amenaza. Para detectar códigos maliciosos, existen dos formas diferentes: basada en firmas (reactiva) o por heurística (proactiva) [14].

  Tres ejemplos de antivirus son ESET [15], Windows Defender [16] y McAffee [17].

- **EDR *(Endpoint Detection and Response)***: es un sistema de protección de los equipos e infraestructuras que identifica vulneraciones de seguridad en tiempo real y desarrolla una respuesta rápida a amenazas potenciales. A diferencia del antivirus, un EDR incorpora detección de amenazas basado en comportamientos, herramientas de análisis apoyadas en el uso del aprendizaje automático, escaneo de IOCs y reglas YARA e interoperabilidad e interacción con otras herramientas de seguridad.

  Trend Micro [18], Kaspersky [19] y SentinelOne [20] son algunos de los EDR que se utilizan actualmente.

- **WAF *(Web Application Firewall)***: es una herramienta especializada en filtrar, monitorizar y bloquear las conexiones desde y hacia una aplicación web. Permiten proteger de ataques de denegación de servicio distribuida, ataques de Cross-Site Scripting (XSS) e inyección SQL, fuerza bruta, correos no deseados, secuestro de sesión, entre otros. Puede desplegarse a nivel de red, a nivel de equipo o en la nube.

  Algunos WAF que existen son Barracuda [21], ModSecurity [22] y Citrix Application Firewall [23].

Estas soluciones permiten tener una visibilidad amplia de lo que sucede en una organización. Sin embargo, revisar estas herramientas una a una resulta ineficiente a la hora de detectar amenazas, por ello, se concentra todas sus actividades en una plataforma SIEM. IBM QRadar [24], Alien Vault [25] y Splunk [26] son tres plataformas SIEM reconocidas en las empresas.

Una vez visto los sistemas que hay para detectar y desplegar en los activos de una empresa, se opta por utilizar un NIDS y un HIDS, además del SIEM.

2.2 Estudio de herramientas

Se va a investigar qué herramientas NIDS y HIDS ofrecen las características necesarias para este proyecto. Respecto a plataforma SIEM, no se va a realizar estudio de las soluciones que hay, ya que se va a trabajar con QRadar.

Se realizan las pruebas en una máquina virtual Ubuntu 22.04, para lo que se toma una instantánea con la máquina virtual limpia y, una vez probada una herramienta, se restaura la máquina para instalar la siguiente.

2.2.1. Network-Based Intrusion Detection System

Respecto a las herramientas NIDS, se va a considerar Snort, Suricata y Zeek.

Snort

Snort es una herramienta desarrollada en lenguaje C y creada por Martin Roesch en 1998. Actualmente, está siendo desarrollada, testeada y aprobada por Cisco. Es *open source* y permite crear a la persona usuaria sus propias reglas de detección, o utilizar aquellas

ya creadas, pudiendo seleccionar para su uso en todo momento las reglas que determine que más se adapten a sus necesidades. La última versión es la 3.1.69.0, liberada el 02 de septiembre de 2023 [28].

Figura 2.1: Logo de Snort

Esta herramienta se puede ejecutar en tres modos diferentes:

- **Sniffer**: muestra los paquetes de red por pantalla.
- **Packet Logger**: almacena los paquetes en el disco.
- **NIDS**: realiza detección y análisis en la red.

Instalación

Se realiza una primera instalación que funciona en modo *sniffer* y *packet logger*, pero no en modo NIDS, por lo que se realiza una segunda instalación que sí permite ejecutar Snort como NIDS.

Se sigue la guía de instalación [27], donde se realizan los siguientes pasos:

1. Se instala las dependencias.
2. Se descarga e instala *Data Acquisition library (DAQ)* que utiliza Snort para hacer llamadas abstractas a bibliotecas de captura de paquetes.
3. Se descarga e instala Snort.

En esta instalación ocurre el siguiente error, en la que falta una fichero cabecera.

```
sp_rpc_check.c:32:10: fatal error: rpc/rpc.h: No such file or directory
   32 | #include <rpc/rpc.h>
      |          ^~~~~~~~~~~
compilation terminated.
make[4]: *** [Makefile:489: sp_rpc_check.o] Error 1
make[4]: Leaving directory '/home/paula/snort_src/snort-2.9.20/src/detection-plu
gins'
make[3]: *** [Makefile:440: all] Error 2
make[3]: Leaving directory '/home/paula/snort_src/snort-2.9.20/src/detection-plu
gins'
make[2]: *** [Makefile:558: all-recursive] Error 1
make[2]: Leaving directory '/home/paula/snort_src/snort-2.9.20/src'
make[1]: *** [Makefile:516: all-recursive] Error 1
make[1]: Leaving directory '/home/paula/snort_src/snort-2.9.20'
make: *** [Makefile:382: all] Error 2
```

Figura 2.2: Error en instalación de Snort

Tras encontrar la solución [29], se explica que los ficheros sí se encuentran pero Snort busca en otro directorio, por lo que hay que copiar el fichero de una carpeta a otra. Se soluciona el error del fichero *rpc.h*, pero hay errores con otros ficheros. Se realiza el siguiente copiado de ficheros, tal y como se muestra en la imagen para solventar los errores.

```
root@pruebas-tfm:/home/paula/snort_src/snort-2.9.20# cp /usr/include/ntirpc/rpc/
*.h /usr/include/rpc/
root@pruebas-tfm:/home/paula/snort_src/snort-2.9.20# cp /usr/include/ntirpc/misc
/*.h /usr/include/misc/
root@pruebas-tfm:/home/paula/snort_src/snort-2.9.20# cp /usr/include/ntirpc/*.h
/usr/include/
```

Figura 2.3: Solución del error en instalación de Snort

Configuración y validación

Se configura Snort para el sistema:

- Se modifica el fichero de configuración (/etc/snort/snort.conf), añadiendo la red que se desea proteger, dónde se encuentran los ficheros de reglas y las listas blanca y negra.
- Se descarga el conjunto de reglas que utilizará Snort. Al tratarse de una prueba, se usarán las reglas de la comunidad.
- Se valida la configuración.

```
MaxRss at the end of detection rules:57860

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.20 GRE (Build 82)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.10.1 (with TPACKET_V3)
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.2.11

           Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 3.2  <Build 1>
           Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
           Preprocessor Object: appid  Version 1.1  <Build 5>
           Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
           Preprocessor Object: SF_S7COMMPLUS  Version 1.0  <Build 1>
           Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
           Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
           Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
           Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
           Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
           Preprocessor Object: SF_POP  Version 1.0  <Build 1>
           Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
           Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
           Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
           Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
           Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>

Total snort Fixed Memory Cost - MaxRss:57860
Snort successfully validated the configuration!
Snort exiting
root@pruebas-tfm:/home/paula/snort_src/snort-2.9.20#
```

Figura 2.4: Validación de la configuración de Snort

Formato de reglas

Las reglas tienen la siguiente estructura:

«acción» «protocolo» «red y puerto» «sentido de la conexión» «red y puerto» «configuración de la regla como el mensaje, el identificador, el contenido del paquete»

Las acciones [30] que se pueden utilizar son:

- **alert**, para generar una alerta;

```
alert tcp $HOME_NET 2589 -> $EXTERNAL_NET any (msg:"MALWARE-BACKDOOR - Dagger_1.4.0";
flow:to_client,established; content:"2|00 00 00 06 00 00 00|Drives|24 00|"; depth:16;
metadata:ruleset community; classtype:misc-activity; sid:105; rev:14;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 7597 (msg:"MALWARE-BACKDOOR QAZ Worm Client Login
access"; flow:to_server,established; content:"qazwsx.hsq"; metadata:ruleset community;
classtype:misc-activity; sid:108; rev:12;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 12345:12346 (msg:"MALWARE-BACKDOOR netbus
getinfo"; flow:to_server,established; content:"GetInfo|0D|"; metadata:ruleset community;
classtype:trojan-activity; sid:110; rev:10;)
```

Figura 2.5: Formato de reglas de Snort

- **block**, para bloquear un paquete;
- **drop**, para eliminar un paquete;
- **log**, para registrar el paquete;
- **pass**, para marcar el paquete como aprobado.

Los protocolos soportados por Snort [31] son:

- IP
- ICMP
- TCP
- UDP

Testeo

Una vez se configura Snort y se conoce el formato de reglas y cuáles son los protocolos y las acciones soportadas, se procede a probar su funcionamiento.

Se crean dos reglas:

- Para detectar conexiones ICMP de cualquier red/IP hacia la red monitoreada.
- Para detecta conexiones a la web de la UPV.

```
root@pruebas-tfm:/home/paula/snort_src# cat /etc/snort/rules/local.rules
alert icmp any any -> $HOME_NET any (msg:"ICMP test"; sid:10000001; rev:001;)
alert tcp any any -> $HOME_NET any (msg:"UPV test"; content:"upv.es"; sid:10000002; rev:001;)
```

Figura 2.6: Reglas de prueba en Snort

Se simula las comunicaciones y Snort las detecta:

Además, se revisan las reglas de la comunidad y reglas propias para familiarizarse con la herramienta.

Integración en QRadar

QRadar dispone de un módulo de soporte para eventos de Snort y se incluye una guía sobre cómo configurar la máquina donde se ejecuta Snort para enviar los registros a QRadar [32] .

```
Commencing packet processing (pid=114767)
05/15-12:25:07.534746  [**] [1:10000002:1] UPV test [**] [Priority: 0] {TCP} 158.42.4.23:80 ->
172.16.215.133:52052
05/15-12:25:07.930955  [**] [1:10000002:1] UPV test [**] [Priority: 0] {TCP} 158.42.4.23:80 ->
172.16.215.133:52054
05/15-12:25:07.951551  [**] [1:10000002:1] UPV test [**] [Priority: 0] {TCP} 158.42.4.23:80 ->
172.16.215.133:52054
05/15-12:25:17.941865  [**] [1:10000001:1] ICMP test [**] [Priority: 0] {ICMP} 172.16.215.1 ->
172.16.215.133
05/15-12:25:17.941889  [**] [1:10000001:1] ICMP test [**] [Priority: 0] {ICMP} 172.16.215.133 -
> 172.16.215.1
05/15-12:25:18.970549  [**] [1:10000001:1] ICMP test [**] [Priority: 0] {ICMP} 172.16.215.1 ->
172.16.215.133
05/15-12:25:18.970570  [**] [1:10000001:1] ICMP test [**] [Priority: 0] {ICMP} 172.16.215.133 -
> 172.16.215.1
```

Figura 2.7: Detección de las reglas en Snort

Suricata

Suricata es un sistema de detección y prevención de intrusiones. Fue desarrollado por la Open Information Security Foundation y está programada en C y Rust. Se lanzó una versión beta en diciembre de 2009, con la primera versión estándar en julio de 2010. La última versión es la 7.0.0, liberada el 18 de julio [33].

Figura 2.8: Logo de Suricata

Suricata utiliza reglas para identificar actividades maliciosas y, lo que distingue a Suricata de otros NIDS, es que tiene una arquitectura de subprocesos múltiples. Con esta arquitectura, Suricata procesa varias tareas de forma simultánea, teniendo un procesamiento de paquetes más rápido y un mejor rendimiento.

Instalación

Se sigue la guía de instalación que hay en la documentación oficial de Suricata [34]. Para Ubuntu, existe un *PPA (Personal Package Archives)* que contiene la última versión estable de Suricata. Se añade el repositorio, se actualiza y se instala.

```
paula@pruebas-tfm:~/Desktop$ suricata -V
This is Suricata version 7.0.0 RELEASE
```

Figura 2.9: Versión de Suricata

Configuración

Para configurar Suricata, es necesario establecer la interfaz y la dirección IP en la que Suricata inspecciona la red de paquetes. El fichero de configuración es */etc/suricata/suricata.yaml* y hay que modificar el valor de la variable *$HOME_NET* y la interfaz.

```
var:
  # more specific is better for alert accuracy and performance
  address-groups:
    HOME_NET: "172.16.215.0/24"
```

Figura 2.10: Fichero de configuración de Suricata

```
# Linux high speed capture support
af-packet:
  - interface: ens33
```

Figura 2.11: Fichero de configuración de Suricata (II)

Se puede personalizar otros valores como la ubicación de los registros, el tipo de alerta que se desea registrar, almacenamiento de ficheros en disco, los ajustes comunes de captura, ajustes de salida.

Reglas

Suricata trabaja con reglas para detectar actividad anómala. Estas reglas tienen un formato similar al de Snort y se almacenan en */var/lib/suricata/rules* según el fichero de configuración, aunque se puede modificar y añadir nuevos ficheros de reglas propios en otras ubicaciones.

```
default-rule-path: /var/lib/suricata/rules

rule-files:
  - suricata.rules
```

Figura 2.12: Fichero de configuración de Suricata (III)

Para que Suricata detecte reglas propias, es necesario crear un fichero *«nombre».rules* donde se almacenarán todas las reglas personalizadas que se necesiten y añadir este fichero de reglas en la configuración de Suricata para que las tenga en cuenta.

Además, Suricata dispone de la herramienta *suricata-update* con la que se pueden obtener, actualizar y gestionar los conjuntos de reglas que se proporcionarán a Suricata. Esta herramienta agrupa todas las reglas en un único fichero, en */var/lib/suricata/rules/suricata.rules*.

```
paula@pruebas-tfm:~$ sudo suricata-update list-sources
24/5/2023 -- 18:12:05 - <Info> -- Using data-directory /var/lib/suricata.
24/5/2023 -- 18:12:05 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
24/5/2023 -- 18:12:05 - <Info> -- Using /usr/share/suricata/rules for Suricata provided rules
24/5/2023 -- 18:12:05 - <Info> -- Found Suricata version 7.0.0 at /usr/bin/suricata.
Name: et/open
  Vendor: Proofpoint
  Summary: Emerging Threats Open Ruleset
  License: MIT
Name: et/pro
  Vendor: Proofpoint
  Summary: Emerging Threats Pro Ruleset
  License: Commercial
  Replaces: et/open
  Parameters: secret-code
  Subscription: https://www.proofpoint.com/us/threat-insight/et-pro-ruleset
Name: oisf/trafficid
  Vendor: OISF
  Summary: Suricata Traffic ID ruleset
  License: MIT
Name: scwx/enhanced
  Vendor: Secureworks
  Summary: Secureworks suricata-enhanced ruleset
  License: Commercial
```

Figura 2.13: Conjunto de reglas de Suricata

Testeo

Se añade el primer conjunto de reglas, el de *Emerging Threats Open Ruleset* de *Proofpoint*.

Se actualiza el conjunto de reglas con *$ suricata-update* y se reinicia el servicio de suricata con *$ systemctl restart suricata*.

```
paula@pruebas-tfm:~$ sudo suricata-update enable-source oisf/trafficid
24/5/2023 -- 18:14:09 - <Info> -- Using data-directory /var/lib/suricata.
24/5/2023 -- 18:14:09 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
24/5/2023 -- 18:14:09 - <Info> -- Using /usr/share/suricata/rules for Suricata provided rules.
24/5/2023 -- 18:14:09 - <Info> -- Found Suricata version 7.0.0 at /usr/bin/suricata.
24/5/2023 -- 18:14:09 - <Info> -- Creating directory /var/lib/suricata/update/sources
24/5/2023 -- 18:14:09 - <Info> -- Enabling default source et/open
24/5/2023 -- 18:14:09 - <Info> -- Source oisf/trafficid enabled
```

Figura 2.14: Adición del conjunto de reglas et/open de Suricata

Para poder comprobar que, realmente, se ha añadido el conjunto de reglas de et/open, Suricata propone la petición al enlace «http://testmynids/index/uid» para generar la alerta *ET POLICY curl User-Agent Outbound*.

```
paula@pruebas-tfm:~$ curl http://testmynids.org/index/uid
<!doctype html>
<html>
    <head>
        <link rel="stylesheet" type="text/css" href="main.css" />
        <title>testmyNIDS.org | tmNIDS.sh</title>
    </head>

    <body>
        <article>
            <h1>Hey! What's up?</h1>
            <div>
                <p>This page is just a placeholder, as this website doesn't have anything worth
 browsing. Its purpose is explained in the project <a href="https://github.com/3CORESec/testmyn
ids.org">Github page</a>. We do <strong>not</strong> host any illegal or malicious content.</p>
                <p>&mdash; <a href="https://twitter.com/3CORESec">@3CORESec</a></p>
            </div>
        </article>
    </body>
</html>
```

Figura 2.15: Petición para generar la alerta ET POLICY curl User-Agent Outbound

```
paula@pruebas-tfm:~$ sudo tail -f /var/log/suricata/fast.log
24/05/2023-18:38:49.973091  [**] [1:2013028:7] ET POLICY curl User-Agent Outbound [**] [Classif
ication: Attempted Information Leak] [Priority: 2] {TCP} 172.16.215.133:36584 -> 18.154.22.96:8
0
```

Figura 2.16: Detección de la alerta ET POLICY curl User-Agent Outbound

Se realiza la prueba con éxito, tal y como se aprecia en las figuras previas. Además, se experimenta con otros conjuntos de reglas y con reglas personalizadas para familiarizarse con la herramienta.

Integración en QRadar

QRadar dispone de módulo de soporte para eventos de Suricata y se incluye las directrices a seguir para enviar los registros de Suricata a QRadar [35].

Zeek

Zeek es un analizador de tráfico de red pasivo y de código abierto, que basa su actividad en el uso de scripts. Vern Paxson comenzó a desarrollar el proyecto en la década de 1990 bajo el nombre de «Bro» como un medio para comprender lo que estaba sucediendo en su universidad y en las redes de laboratorios nacionales. Vern y el equipo de liderazgo del proyecto cambiaron el nombre de «Bro» a «Zeek» a finales de 2018 para celebrar su expansión y desarrollo continuo. La última versión es la 6.0.0, liberada el 31 de mayo [36].

Figura 2.17: Logo de Zeek

Instalación

Para instalar Zeek:

1. Se instalan las dependencias que se indican en la documentación oficial [37].
2. Se sigue la guía de instalación de GitHub [38]. Para la instalación, tarda alrededor de 2 horas.

Figura 2.18: Versión de Zeek

Configuración y validación

Para configurar Zeek, hay que conocer los dos modos de configuración que existen y seleccionar el que mejor convenga:

- *Modo standalone*: en la que Zeek se ejecuta en la propia máquina. Es conveniente cuando no hay mucho tráfico de red.
- *Modo clúster*: este modo se suele utilizar cuando el volumen del tráfico es elevado para ser analizado por un único proceso y se dividen el trabajo en múltiples nodos. Los cuatro componentes principales que participan en este modo son [39]:
  - *Manager*: recibe los mensajes de registros y avisos de los otros nodos, y los combina con los registros que se producen en los nodos trabajadores (*workers*).
  - *Worker*: realiza la captación de los paquetes que transitan por la red y realiza el análisis.
  - *Proxy*: puede utilizarse para descargar el almacenamiento de datos o cualquier carga de trabajo arbitraria.
  - *Logger*: es opcional y puede utilizarse para recibir los mensajes de registros de otros nodos, para reducir la carga del nodo *manager*.

En este caso, dado que el objetivo es familiarizarse con la herramienta y no se pretende generar un gran volumen de tráfico, se prueba con el modo *standalone*.

Los cambios necesarios que hay que hacer en el fichero de configuración de Zeek, ubicado en «*/usr/local/zeek/etc/node.cfg*», consisten en:

- Especificar la interfaz a monitorizar.
- Establecer el modo de funcionamiento, *standalone* en este caso.

```
paula@pruebas-tfm:~/Desktop/zeek$ sudo cat /usr/local/zeek/etc/node.cfg
# This is a complete standalone configuration.  Most likely you will
# only need to change the interface.
[zeek]
type=standalone
host=localhost
interface=ens33
```

Figura 2.19: Fichero de configuración *node.cfg*

ZeekControl es una herramienta que facilita la gestión de Zeek. La primera vez que se utiliza, es necesario realizar una instalación inicial de la configuración, para lo que se accede al entorno de ZeekControl con *zeekctl* y, una vez dentro, se utiliza el comando *install*.

```
paula@pruebas-tfm:~/Desktop/zeek$ zeekctl
Warning: zeekctl config has changed (run the zeekctl "deploy" command)

Welcome to ZeekControl 2.5.0-24

Type "help" for help.

[ZeekControl] > install
removing old policies in /opt/zeek/spool/installed-scripts-do-not-touch/site ...
removing old policies in /opt/zeek/spool/installed-scripts-do-not-touch/auto ...
creating policy directories ...
installing site policies ...
generating standalone-layout.zeek ...
generating local-networks.zeek ...
generating zeekctl-config.zeek ...
generating zeekctl-config.sh ...
```

Figura 2.20: Instalación de configuración de ZeekControl

Una vez realizada la instalación, se comprueba con el comando *check* que el fichero de configuración modificado anteriormente no contiene errores y se procede a ejecutar una instancia de Zeek con los nuevos cambios con *deploy*.

```
[ZeekControl] > check
zeek scripts are ok.
[ZeekControl] > deploy
checking configurations ...
installing ...
removing old policies in /opt/zeek/spool/installed-scripts-do-not-touch/site ...
removing old policies in /opt/zeek/spool/installed-scripts-do-not-touch/auto ...
creating policy directories ...
installing site policies ...
generating standalone-layout.zeek ...
generating local-networks.zeek ...
generating zeekctl-config.zeek ...
generating zeekctl-config.sh ...
stopping ...
stopping zeek ...
starting ...
starting zeek ...
```

Figura 2.21: Comando *check* y *deploy* de ZeekControl

Scripts

Zeek no es un sistema clásico de detección de intrusiones basado en firmas o reglas, aunque sí es compatible con ellas. El uso de scripts ofrece un espectro mucho más amplio de enfoques muy diferentes para encontrar actividades maliciosas. Entre ellos se incluyen la detección semántica de usos indebidos, la detección de anomalías y el análisis del comportamiento.

Estos scripts se encuentran en la carpeta */usr/local/zeek/share/zeek*:

- *base*: contiene los scripts que siempre carga Zeek y se encargan de recopilar información sobre las actividades de red o de proporcionar utilidades que mejoran la funcionalidad de Zeek. No se deben modificar.

- *policy*: guarda todo los scripts de políticas adicionales y no se debe modificar. Zeek carga algunos de estos scripts, pero es la persona usuaria la que elige cuál desea cargar.
- *site*: donde se almacenan los scripts personalizados.

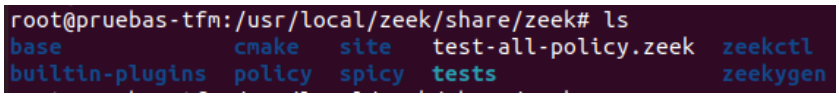


Figura 2.22: Contenido de */usr/local/zeek/share/zeek*

Al acceder a la carpeta *base*, se encuentran algunos scripts de Zeek (.zeek), para iniciar el servicio, y algunas carpetas como *protocols* y *utils*.

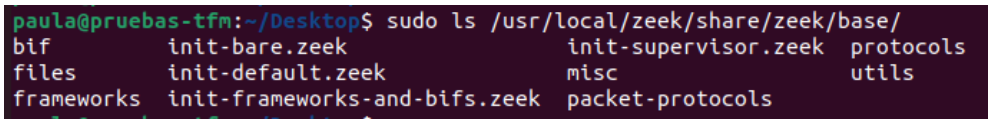


Figura 2.23: Contenido de */usr/local/zeek/share/zeek/base*

En *protocols*, hay múltiples carpetas con todos los protocolos que se pueden detectar:

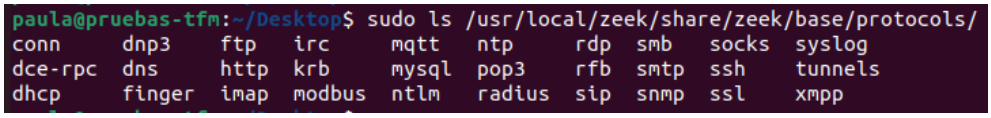


Figura 2.24: Contenido de */usr/local/zeek/share/zeek/base/protocols*

Se accede a la carpeta de HTTP y se observa que hay múltiples scripts, como *entities.zeek*, *files.zeek* y *main.zeek*. La ejecución de *__load__.zeek* consiste en cargar los otros ficheros.

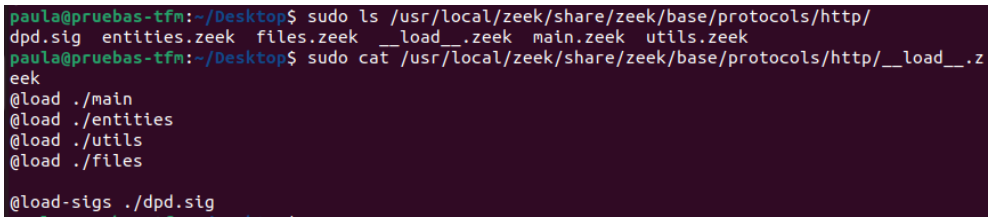


Figura 2.25: Scripts para monitorizar tráfico HTTP

Se comprueba que ocurre lo mismo con otros protocolos, por lo que se aprecia la estructura de los scripts de Zeek y cómo se componen.

Se abre el contenido de *utils.zeek* y se compone de funciones para procesar HTTP. Estas funciones se encargan de extraer claves y construir URLs.

```
function extract_keys(data: string, kv_splitter: pattern): string_vec
        {
        local key_vec: vector of string = vector();

        local parts = split_string(data, kv_splitter);
        for ( part_index in parts )
                {
                local key_val = split_string1(parts[part_index], /=/);
                if ( 0 in key_val )
                        key_vec += key_val[0];
                }
        return key_vec;
        }
```

Figura 2.26: Función *extract_keys* del script *utils.zeek* de HTTP

```
function build_url(rec: Info): string
        {
        local uri  = rec?$uri ? rec$uri : "/<missed_request>";
        if ( strstr(uri, "://") != 0 )
                return uri;

        local host = rec?$host ? rec$host : addr_to_uri(rec$id$resp_h);
        local resp_p = port_to_count(rec$id$resp_p);
        if ( resp_p != 80 )
                host = fmt("%s:%d", host, resp_p);
        return fmt("%s%s", host, uri);
        }

function build_url_http(rec: Info): string
        {
        return fmt("http://%s", build_url(rec));
        }

function describe(rec: Info): string
        {
        return build_url_http(rec);
        }
```

Figura 2.27: Funciones para URLs del script *utils.zeek* de HTTP

Con todo lo visto anteriormente, se puede concluir que Zeek utiliza un lenguaje y estructura propios. Para poder habituarse a los scripts, Zeek proporciona un tutorial interactivo [40].

Ficheros de registros

Además de los ficheros de registro de protocolos de red convencional (como son http.log, dns.log, ssh.log, entre otros), Zeek genera otros ficheros de registros importantes basados en estadísticas y actividad interesante que observa en el tráfico. Algunos de estos ficheros son:

- *conn.log*: almacena todas las conexiones que se han registrado con la mayor información que ha podido obtener de ellas, como las IPs origen y destino, servicios, puertos, tamaño.
- *notice.log*: identifica actividad específica que Zeek reconoce como extraña, anómala o interesante.
- *known_services.log*: contiene los servicios detectados en la red local y que se sabe que son utilizados activamente por los clientes de la red.
- *weird.log*: registra actividad inusual o excepcional como puede ser mal funcionamiento o configuración, tráfico que no es habitual a un protocolo.

Aunque estos ficheros resultan interesantes, no se prueban, ya que se requiere de un volumen considerable de tráfico que no se le proporciona en el testeo de esta herramienta.

Los registros de protocolos de red sí se comprueban cómo se almacena y qué campos recoge. En la siguiente figura, se aprecia las conexiones HTTP que se han simulado.


```
{"ts":1690626839.012969,"uid":"C7OXe91hZnsI6jF8M3","id.orig_h":"172.16.215.133","id.orig_p":52
762,"id.resp_h":"142.250.184.3","id.resp_p":80,"trans_depth":1,"version":"1.1","request_body_l
en":0,"response_body_len":471,"status_code":200,"status_msg":"OK","tags":[],"resp_fuids":["FMO
RACBfqr2rxK842"],"resp_mime_types":["application/ocsp-response"]}
{"ts":1690626839.184219,"uid":"CJQCdA1mgw7aUS8BFb","id.orig_h":"172.16.215.133","id.orig_p":52
752,"id.resp_h":"142.250.184.3","id.resp_p":80,"trans_depth":2,"version":"1.1","request_body_l
en":0,"response_body_len":471,"status_code":200,"status_msg":"OK","tags":[],"resp_fuids":["FQ8
xWQ3YY6JJJik8vk"],"resp_mime_types":["application/ocsp-response"]}
{"ts":1690626867.522334,"uid":"CKd0mE2DAbVKOq5BO7","id.orig_h":"172.16.215.133","id.orig_p":54
078,"id.resp_h":"185.125.190.48","id.resp_p":80,"trans_depth":1,"version":"1.1","request_body_
len":0,"response_body_len":0,"status_code":204,"status_msg":"No Content","tags":[]}
{ts:1690626867.522334,uid:CKd0mE2DAbVKOq5BO7,id.orig_h:172.16.215.133,id.orig_p:54078,id.resp_
h:185.125.190.48,id.resp_p:80,trans_depth:1,version:1.1,request_body_len:0,response_body_len:0
,status_code:204,status_msg:No Content,tags:[]}
{ts:1690626867.522334,uid:CKd0mE2DAbVKOq5BO7,id.orig_h:172.16.215.133,id.orig_p:54078,id.resp_
h:185.125.190.48,id.resp_p:80,trans_depth:1,version:1.1,request_body_len:0,response_body_len:0
,status_code:204,status_msg:No Content,tags:[]}
{ts:1690626867.522334,uid:CKd0mE2DAbVKOq5BO7,id.orig_h:172.16.215.133,id.orig_p:54078,id.resp_
h:185.125.190.48,id.resp_p:80,trans_depth:1,version:1.1,request_body_len:0,response_body_len:0
,status_code:204,status_msg:No Content,tags:[]}
```


Figura 2.28: Fichero *http.json*

Muestra campos como las IPs origen y destino, los puertos origen y destino, la respuesta, el tamaño de la respuesta, entre otros datos.

Integración en QRadar

QRadar no dispone de módulo de soporte para eventos de Zeek ni incluye directrices para enviar los registros de Zeek a QRadar.

2.2.2. Host-Based Intrusion Detection System

Respecto a las herramientas HIDS, se va a valorar OSSEC y Samhain.

OSSEC

OSSEC es un proyecto en software libre que se puede adaptar a los requerimientos de seguridad a través de la detección de intrusos. Además, analiza los registros de eventos del sistema operativo, comprueba la integridad del mismo, auditorías de los registros, detecta de rootkits, alertas en tiempo real y respuesta activa a ataques. La última versión es la 3.7.0, que se liberó el 17 de enero del 2022 [41].

Figura 2.29: Logo de OSSEC

Instalación

Para la instalación de OSSEC:

- Se instala los paquetes requeridos para Ubuntu según la documentación de OSSEC [42].
- Se descarga el paquete de la página oficial [43] y se sigue los pasos para la instalación [44].

Durante la instalación, hay el siguiente error en el que el enlazador no encuentra *lsystemd*.

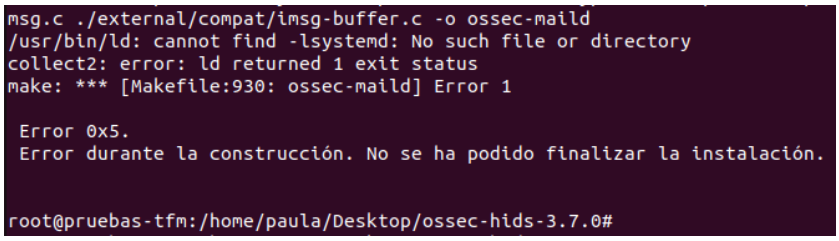


Figura 2.30: Error en la instalación de OSSEC

La solución se encuentra en el propio GitHub de OSSEC [45], en el que se explica que hay que instalar el paquete *libsystemd-dev*.

Tras solucionar el error, ya se puede continuar con la instalación. Al ejecutar *./install.sh*, solicita el idioma.

Figura 2.31: Instalación OSSEC

Una vez seleccionado el idioma, realiza una serie de preguntas para poder instalar y configurar OSSEC. Esta herramienta puede ejecutarse en modo cliente-servidor o en modo local. En este caso, se ha optado por una instalación local, con servidor de integridad del sistema y sistema de detección de rootkit.

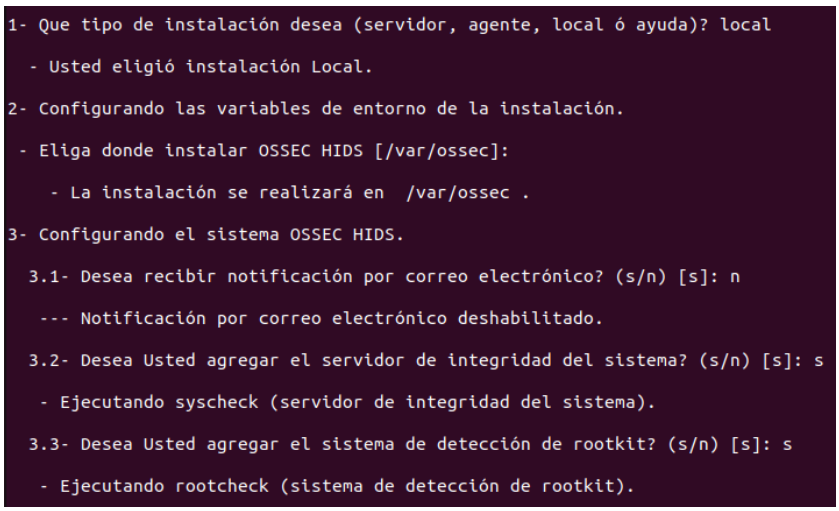


Figura 2.32: Instalación OSSEC (II)

Configuración

El fichero de configuración es */var/ossec/etc/ossec.conf*. En dicho archivo, están las reglas que utiliza OSSEC, los directorios a ignorar, el nivel de las alertas a registrar, entre otra información. Todos estos parámetros se pueden personalizar.

```
root@pruebas-tfm:~# cat /var/ossec/etc/ossec.conf
<ossec_config>
  <global>
    <email_notification>no</email_notification>
  </global>

  <rules>
    <include>rules_config.xml</include>
    <include>pam_rules.xml</include>
    <include>sshd_rules.xml</include>
    <include>telnetd_rules.xml</include>
    <include>syslog_rules.xml</include>
```

Figura 2.33: Fichero de configuración de OSSEC

En la propia instalación, ya se configura una parte de estos parámetros como la respuesta activa, la notificación por correo y la detección de rootkit.

Reglas

Las reglas se encuentran en */var/ossec/rules/* y están en formato XML.

En la figura se muestran dos reglas del fichero *policy_rules.xml*. En ellas, se detecta accesos satisfactorios en horario no laboral y en fines de semana.

```
root@pruebas-tfm:~# tail -n 20 /var/ossec/rules/policy_rules.xml

<group name="policy_violation,">
  <rule id="17101" level="9">
    <if_group>authentication_success</if_group>
    <time>6 pm - 8:30 am</time>
    <description>Successful login during non-business hours.</description>
    <group>login_time,</group>
  </rule>

  <rule id="17102" level="9">
    <if_group>authentication_success</if_group>
    <weekday>weekends</weekday>
    <description>Successful login during weekend.</description>
    <group>login_day,</group>
  </rule>
</group> <!-- POLICY_RULES -->


<!-- EOF -->
```

Figura 2.34: Fichero *policy_rules.xml*

A cada regla se le asigna un identificador (*id*), un nivel (*level*) y una descripción (*description*). En la primera regla, se añade la condición de si pertenece al grupo de autenticación satisfactoria y la hora en el que sucede el evento es de 6 de la tarde a 8 de la mañana. En la segunda regla, se establece como condición los días de semana.

Testeo

OSSEC dispone de ejecutables para gestionar la herramienta, que están en */var/ossec/-bin/*. A continuación, se muestra los archivos ejecutables que hay en dicha carpeta.

```
root@pruebas-tfm:/home/paula/Desktop# ls /var/ossec/bin/
agent_control       ossec-dbd           ossec-reportd
clear_stats         ossec-execd         ossec-syscheckd
list_agents         ossec-logcollector  rootcheck_control
manage_agents       ossec-logtest       syscheck_control
ossec-agentlessd    ossec-maild         syscheck_update
ossec-analysisd     ossec-makelists     util.sh
ossec-authd         ossec-monitord      verify-agent-conf
ossec-control       ossec-regex
ossec-csyslogd      ossec-remoted
```

Figura 2.35: Contenido de */var/ossec/bin/*

Estos archivos permiten la gestión de los agentes, comprobar y verificar la configuración y ejecutar o parar el servicio.

Para comenzar a monitorizar, se utiliza el comando */var/ossec/bin/ossec-control start*:

```
paula@pruebas-tfm:~/Desktop$ sudo /var/ossec/bin/ossec-control start
Starting OSSEC HIDS v3.7.0...
2023/07/02 17:31:13 ossec-maild: INFO: E-Mail notification disabled. Clean Exit.
Started ossec-maild...
Started ossec-execd...
ossec-analysisd already running...
ossec-logcollector already running...
ossec-syscheckd already running...
ossec-monitord already running...
Completed.
```

Figura 2.36: Comando para ejecutar OSSEC

Las alertas se almacenan en el registro */var/ossec/logs/alerts/alerts.log*. Se ejecuta el comando para ver el registro de alertas con sudo y en otra terminal se escala privilegios a root, lo que genera las alertas que se muestran en la figura:

```
paula@pruebas-tfm:~$ sudo tail -f /var/ossec/logs/alerts/alerts.log
** Alert 1688312184.23743: - pam,syslog,authentication_success,
2023 Jul 02 17:36:24 pruebas-tfm->/var/log/auth.log
Rule: 5501 (level 3) -> 'Login session opened.'
Jul  2 17:36:23 pruebas-tfm sudo: pam_unix(sudo:session): session opened for use
r root(uid=0) by (uid=1000)

** Alert 1688312184.24016: - pam,syslog,
2023 Jul 02 17:36:24 pruebas-tfm->/var/log/auth.log
Rule: 5502 (level 3) -> 'Login session closed.'
Jul  2 17:36:24 pruebas-tfm sudo: pam_unix(sudo:session): session closed for use
r root

** Alert 1688312186.24245: - syslog,sudo
2023 Jul 02 17:36:26 pruebas-tfm->/var/log/auth.log
Rule: 5402 (level 3) -> 'Successful sudo to ROOT executed'
User: paula
Jul  2 17:36:25 pruebas-tfm sudo:    paula : TTY=pts/0 ; PWD=/home/paula ; USER=
root ; COMMAND=/usr/bin/tail -f /var/ossec/logs/alerts/alerts.log
```

Figura 2.37: Eventos detectados por OSSEC

Integración en QRadar

QRadar dispone de módulo de soporte para eventos de OSSEC y se incluye las directrices a seguir para enviar las alertas detectadas por OSSEC a QRadar [46].

Samhain

Samhain es el sistema de detección de intrusos basado en host que proporciona verificación de integridad del archivo y monitoreo, análisis de archivos de registro, así como detección de rootkit, detección de procesos ocultos. La última versión es la 4.4.10, liberada el 14 de mayo de 2023.

Figura 2.38: Logo de Samhain

Samhain se trata de una solución alemana que ha sido desarrollado por *Samhain Labs*. Otro de sus proyectos, Beltane, consiste en una consola de gestión centralizada basada en la web para los sistemas Samhain.

Instalación

Para la instalación, se sigue la guía que se proporciona en la web de Samhain [47].

Al realizar la instalación, aparece un mensaje de error en el que se especifica que la ruta */var/log* no es una ruta de confianza, ya que pertenece al miembro *syslog* y Samhain no reconoce a este miembro como uno de confianza.

```
trustfile: group writeable, group_gid: 111
trustfile: checking group member syslog, uid 104
trustfile: uid=104, trusted_uid=0, no match
trustfile: uid=104, trusted_uid=111, no match
trustfile: uid=104, trusted_uid=0, no match
trustfile: group member syslog not found in trusted users --> ERROR
-------------------------------------------
trustfile: EBADGID 111 /var/log (group member not trusted)
This file/directory is group writeable, and one of the group members
is not in samhains list of trusted users.
Please run ./configure again with the option
 ./configure [more options] --with-trusted=0,...,UID
where UID is the UID of the (yet) untrusted user.
-------------------------------------------
trustfile: ERROR: not a trusted path: /var/log
```

Figura 2.39: Error al instalar Samhain

Los usuarios de confianza son *root* y el usuario efectivo del proceso. Se puede configurar miembros de confianza adicionales de dos formas diferentes:

- En tiempo de compilación, con la opción *./configure –with-trusted=0,...*.
- Añadiendo los miembros en el fichero de configuración, en la variable *TrustedUsers*.

Se ha prueba ambas opciones con diferentes formas, pasando el UID, el nombre del usuario, y no se puede solventar el error. Se intenta ejecutar, pero la herramienta necesita el acceso a la carpeta */var/log* para su funcionamiento, por tanto, no se puede testear su actividad.

Integración en QRadar

QRadar dispone de un módulo de soporte para eventos de Samhain y se incluye las directrices a seguir para enviar las alertas detectadas por Samhain a QRadar [48].

2.3 Selección de herramientas

Se realiza la comparación de las herramientas NIDS y HIDS que se han visto previamente en base a ciertos aspectos comunes y esenciales, como la facilidad de instalación y configuración, el soporte que hay por parte de la comunidad y cuándo fue la última actualización, entre otros.

2.3.1. Network-Based Intrusion Detection System

Snort es una herramienta que no es difícil de instalar y configurar que permite detectar tráfico en tiempo real, está basado en reglas. La última actualización es del 02 de septiembre, siendo la versión actual la 3.1.69.0; la actualización anterior a esta es de agosto, con lo que se comprueba que está en continuo crecimiento, además tiene un buen soporte por parte de la comunidad. En cuanto a protocolos, está un poco limitado, ya que solo detecta TCP, UDP, IP e ICMP, no permite la detección del protocolo independientemente del puerto y tampoco admite la extracción de archivos ni puede ejecutarse con subprocesos múltiples.

Suricata es una herramienta sencilla de instalar y configurar, trabaja con reglas y scripts y en tiempo real. La versión actual es la 7.0.0, lanzada en julio, y la actualización anterior a esta fue en junio, verificando que esta herramienta se encuentra en constante desarrollo. Tiene un buen soporte, con una buena documentación en su web así como en foros y buena información que aporta la comunidad. Permite detectar una gran variedad de protocolos, entre los que están TCP, UDP, IP, ICMP, HTTP, SMTP, POP3, TLS/SSL, SMB, entre otros. Asimismo, admite la extracción de archivos, el análisis de malware y la detección del protocolo independiente del puerto. Una característica a destacar de Suricata es que es un programa de subprocesos múltiples, por lo que habrá varios subprocesos trabajando al mismo tiempo, mejorando el rendimiento.

Zeek, por su parte, es una herramienta con una instalación lenta y que tiene cierta complejidad a la hora de configurar. Trabaja con scripts en un lenguaje propio y en tiempo real. La versión actual de la herramienta es la 6.0.0 que fue lanzada en mayo, con lo que se confirma que se encuentra mantenida, aunque tiene un gran soporte por parte de la comunidad. Al igual que Suricata, analiza un gran abanico de protocolos, entre los que se encuentran DHCP, DNS, FTP, HTTP, SMTP, SSL, y se ejecuta con subprocesos múltiples. Además, permite la extracción de archivos y el análisis de malware, pero no detecta el protocolo independientemente del puerto.

	Snort	**Suricata**	**Zeek**
Instalación	Normal y rápida	Sencilla y rápida	Normal y lenta
Configuración	Sencilla	Sencilla	No sencilla
Versión actual	3.1.69.0	7.0.0	6.0.0
Fecha de liberación	02/09/23	18/07/23	31/05/23
Comunidad	Buena	Buena	Mejorable
Integración en QRadar	Soportada	Soportada	No soportada
Protocolos	TCP, UDP, IP, ICMP	TCP, UDP, IP, ICMP, HTTP, SMTP...	TCP, UDP, IP DHCP, DNS, FTP, HTTP, SMTP...
Extracción de archivos	No soportado	Soportado	Soportado
Multithreading	No soportado	Soportado	Soportado

Tabla 2.1: Comparación entre herramientas NIDS

Tras esta comparativa, donde se han equiparado las herramientas según ciertos criterios, y tras haber trabajado con las tres soluciones, se ha optado utilizar Suricata por varias razones.

Aunque Snort y Suricata coinciden en algunos aspectos, como la facilidad de instalación y configuración, y ambas se encuentran en constante desarrollo, Suricata ofrece un rango de protocolos bastante mayor que Snort, además de la extracción de archivos, el *multithreading* y la detección del protocolo independiente del puerto.

Zeek, por su parte, coincide con Suricata en lo que aportan con la extracción de archivos y la gran variedad de protocolos, pero consumen un tiempo considerable en la instalación y no es sencilla de configurar. Aunque Zeek ofrece unas propiedades muy interesantes, requiere de una dedicación y tiempos excesivos. Asimismo, no hay un parseador de Zeek ni una documentación para integrarlo en QRadar.

2.3.2. Host-Based Intrusion Detection System

OSSEC es una herramienta que es sencilla y rápida de instalar y configurar, además de ser intuitiva. Trabaja con ficheros de reglas en formato XML y se puede gestionar la aplicación con sus propios scripts. La versión actual de la herramienta, que es la 3.7.0, fue lanzada en enero de 2022, y la anterior a esta fue en febrero de 2020, con lo que se encuentra mantenida pero no se encuentra muy actualizada. Tiene un gran soporte por parte de la comunidad. Además, la experiencia tratando con la herramienta ha sido positiva.

Samhain es sencilla de instalar. La configuración también parece sencilla, ya que se puede realizar por parámetros en tiempo de compilación o modificando el fichero de configuración. Presenta unas buenas características que no se han podido examinar. Se trata de un proyecto que se encuentra en constante desarrollo, ya que la última versión, que es la 4.4.10, se liberó hace cuatro meses. La documentación que ofrecen es básica y podría mejorarse, así como el soporte que tiene por parte de la comunidad.

	OSSEC	**Samhain**
Instalación	Sencilla y rápida	Sencilla y rápida
Configuración	Sencilla	Sencilla
Versión actual	3.7.0	4.4.10
Fecha de liberación	17/01/22	14/05/23
Comunidad	Buena	Mejorable
Integración en QRadar	Soportada	Soportada

Tabla 2.2: Comparación entre herramientas HIDS

Aunque Samhain se encuentra más actualizada y tiene unas propiedades interesantes, no se ha podido probar su funcionamiento, por lo que no se ha podido realizar una comparación exhaustiva con OSSEC.

Tras el estudio de la herramienta y la comparativa se opta por utilizar OSSEC como HIDS para este proyecto.

CAPÍTULO 3
Implementación

En este capítulo, se procede con la implementación de la solución con QRadar como plataforma SIEM, Suricata como NIDS y OSSEC como HIDS.

Para la simulación del escenario, se utilizan tres máquinas virtuales que se disponen de la siguiente forma:

- **QRadar**: una máquina virtual con CentOS 7.7 que contiene QRadar.
- **Cliente**: Una máquina virtual con Ubuntu 22.04 simulará la actividad del cliente y tendrá instalado Suricata y OSSEC como agente; esta es la máquina que será monitorizada.
- **OSSEC**: Una máquina virtual con Ubuntu 22.04 que contendrá el servidor OSSEC que gestiona los eventos y alertas de la máquina del cliente.

Para la gestión y ejecución de máquinas virtuales se utiliza VMWare.

Una vez Suricata y OSSEC se encuentren instaladas y configuradas, se procede a su integración en QRadar así como al mapeo de eventos y creación de reglas.

3.1 QRadar

IBM pone a disposición una versión gratuita de QRadar, conocida como «*IBM QRadar Community Edition V7.3.3*» [49], para permitir a las personas usuarias, estudiantes y profesionales de seguridad aprender y experimentar con QRadar.

Community Edition tiene una licencia perpertua, aunque está limitada para un uso no empresarial, por lo que solo permite 50 eventos por segundo y 5.000 flujos de red por minuto, entre otras restricciones.

3.1.1. Requisitos

Los requisitos mínimos que necesita *QRadar Community Edition* para funcionar correctamente son los siguientes:

- Memoria mínima de 8GB de RAM.
- Espacio en disco mínimo de 250GB.
- CPU: 2 núcleos (mínimo) o 6 núcleos (recomendado).

- Se requiere un adaptador de red con acceso a Internet.
- Se requiere una dirección IP estática pública y privada para QRadar Community Edition.
- El nombre de host asignado debe ser un nombre de dominio completo.

Estos requerimientos se tendrán en cuenta a la hora de crear la máquina virtual.

3.1.2. Descarga e instalación

IBM ofrece un fichero OVA con *QRadar Community Edition* [50] para facilitar la ejecución de QRadar. Se descarga el fichero y en VMware, se crea una máquina virtual con la OVA. Se establece la configuración de la máquina virtual para satisfacer los requisitos:

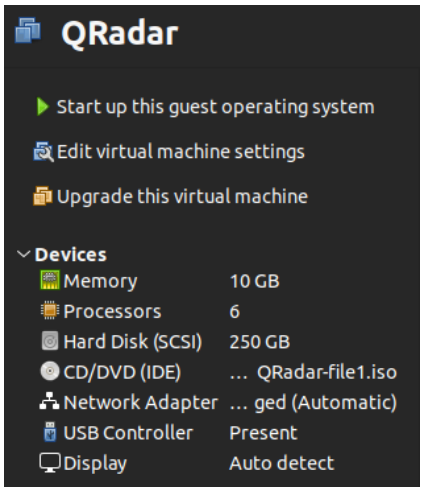


Figura 3.1: Configuración de máquina virtual QRadar

Para la instalación, se ha seguido el manual de QRadar ofrecido por IBM [51].

Al acceder a la máquina virtual por primera vez, se inicia sesión con *root* y se solicita el cambio de contraseña.

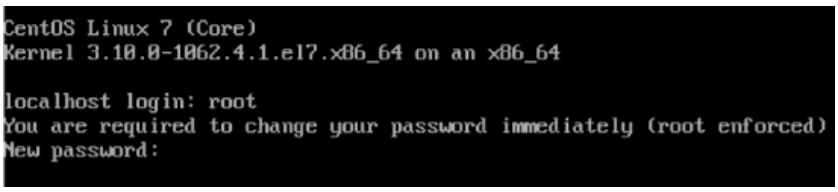


Figura 3.2: Acceso con root y cambio de contraseña

Tras la solicitud de cambio de contraseña, se procede a la instalación con *./setup*. Se acepta la EULA y se confirma la instalación. El proceso de instalación dura, aproximadamente, una hora. Una vez instalado, se gestiona la cuenta de *admin*, cambiando la contraseña. Esta cuenta de *admin* es necesaria para acceder a la consola por web.

La IP de la máquina de QRadar es la *192.168.1.36*. Se accede en el navegador con esta IP y se muestra la página de inicio de sesión. Las credenciales para acceder son la cuenta *admin* y la contraseña establecida anteriormente. Se muestra un acuerdo de licencia que hay que aceptar.

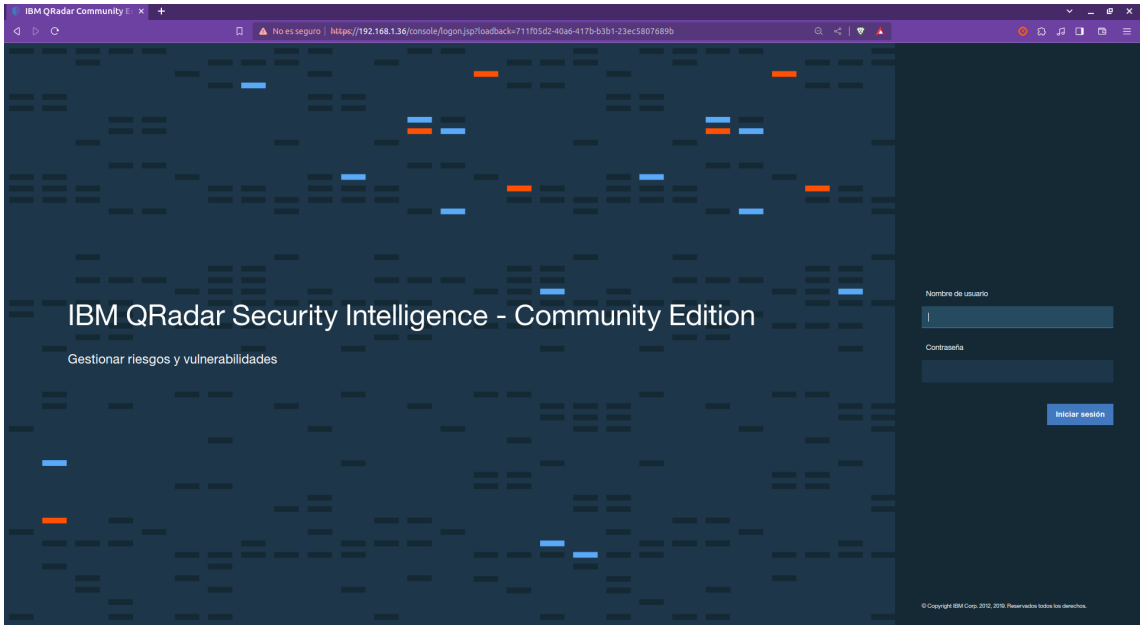


Figura 3.3: Página de inicio de sesión a QRadar

Al acceder por primera vez a la consola web, es necesario reiniciar el sistema. En la pestaña «Admin», se accede en *System Configuration* a *System and License Management*. En la nueva ventana que se abre, se clica en el host y se reinicia el sistema.

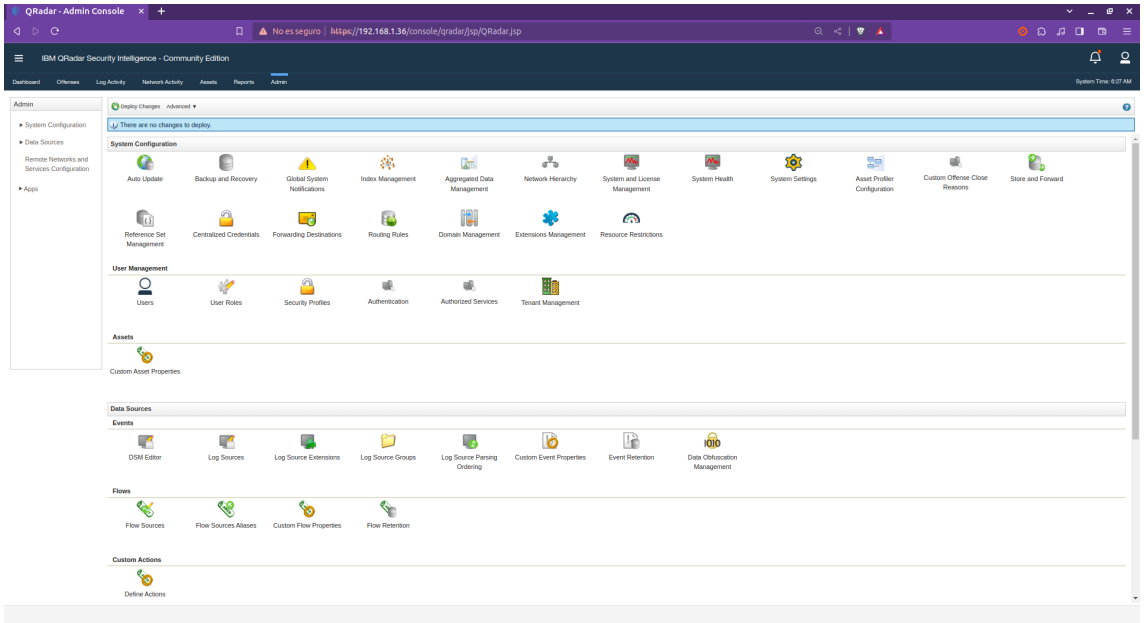


Figura 3.4: Pestaña admin

Este proceso tarda alrededor de 20 minutos.

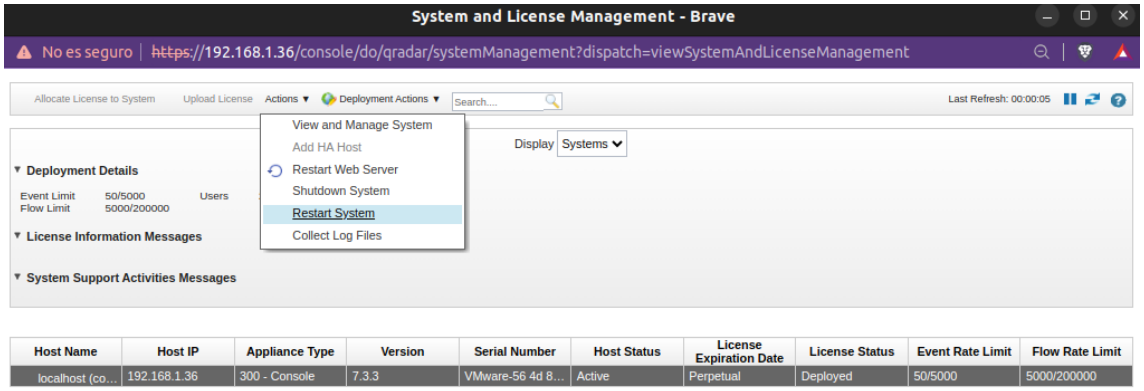


Figura 3.5: Reiniciar sistema

3.1.3. Dificultades encontradas

Corrección de licencia

Se decide probar con la fuente de *Squid*, para verificar que QRadar funciona correctamente. Sin embargo, se descubre que no recibe los eventos como debería. Tras buscar, se descubre que IBM actualizó unas instrucciones en QRadar para aplicar una corrección de licencia a los dispositivos *Disconnected Log Collector* [53]. Ejecutando el comando para *Community Edition*, ya se reciben los registros.

Acceso a la consola web

Cuando se inicia la máquina de QRadar, hay que esperar a que todos los servicios se activen. Si los servicios no se están ejecutando, no se permite la conexión a la consola web o el acceso es muy lento.

Las complicaciones más comunes son que la máquina rechaza la conexión o, si muestra el portal de acceso, no logra acceder y se queda esperando durante más de 10 minutos.

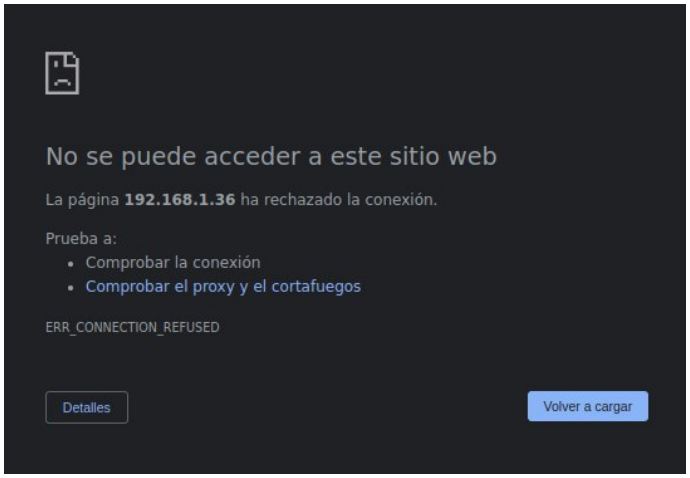


Figura 3.6: Rechazo en la conexión

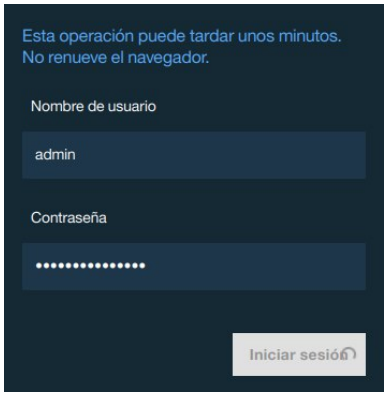


Figura 3.7: Error en el acceso

La solución que se encuentra es la de reiniciar los servicios de QRadar [54], los cuales son *Host service*, *Tomcat*, *Host Context* y *Network Manager*. El proceso de parar y comenzar estos procesos no es rápido, requiere de tiempo.

Al comprobar que esto ocurre cada vez que se accede a la máquina, se decide crear un script para comprobar la IP de la máquina y el estado de los servicios.

```
[root@localhost ~]# ./estado-maquina.sh

 ----- IP -----
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc prio state UP group default qlen 1000
    inet 192.168.1.36/24 brd 192.168.1.255 scope global ens33

 ----- NETWORK MANAGER -----
Redirecting to /bin/systemctl status NetworkManager.service
   Active: active (running) since Mon 2023-09-11 06:19:03 UTC; 56min ago

 ----- HOST SERVICES -----
Redirecting to /bin/systemctl status hostservices.service
   Active: active (exited) since Mon 2023-09-11 06:17:41 UTC; 58min ago

 ----- TOMCAT -----
Redirecting to /bin/systemctl status tomcat.service
   Active: active (running) since Mon 2023-09-11 06:18:12 UTC; 57min ago

 ----- HOST CONTEXT -----
Redirecting to /bin/systemctl status hostcontext.service
   Active: active (running) since Mon 2023-09-11 06:18:12 UTC; 57min ago
[root@localhost ~]#
```

Figura 3.8: Script para comprobar el estado

Los servicios se van iniciando y suelen requerir de entre 30 segundos a 90 segundos hasta encontrarse todos activos. Sin embargo, en la mayor parte de las ocasiones, el servicio de *Network Manager* no se inicia solo, hay que ejecutarlo manualmente con *systemctl*. Por ello, se crea el siguiente script.

```
[root@localhost ~]# cat ./reiniciar-nm.sh
systemctl start NetworkManager
[root@localhost ~]#
```

Figura 3.9: Script para reiniciar el servicio de NetworkManager

Cada vez que se inicia la máquina de QRadar, se comprueba el estado de los servicios y se inicia el proceso de NetworkManager.

Conexión al servidor de consulta

Al acceder a la pestaña de *Log Activity* para visualizar los registros, aparece el mensaje de error «*There was a problem connecting to the query server. Please try again later*».

Este error está relacionado con los servicios de Ariel Server [55].

3.2 Suricata

La máquina en la que se instala Suricata es la máquina que simula la actividad del cliente. Se trata de una Ubuntu 22.04 con IP *192.168.1.38* e interfaz *ens33*.

Es necesario que la IP de esta máquina permanezca estática. Para ello, se modifica el fichero de configuración *01-network-manager-all.yaml* de netplan y se reinicia el servicio.

```
root@cliente:/home/cliente/Desktop# cat /etc/netplan/01-network-manager-all.yaml

# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    ens33:
      dhcp4: no
      addresses: [192.168.1.38/24]
      gateway4: 192.168.1.1
      nameservers:
         addresses: [8.8.8.8,8.8.8.4]
```

Figura 3.10: Configuración para IP estática

Una vez configurada la IP estática, se realiza la instalación de Suricata con el *PPA*, como se indica en el apartado de Suricata en la sección 2.2.1.

3.2.1. Configuración

El primer paso de la configuración consiste en informar a Suricata sobre la red a monitorizar, donde se especifica la variable *HOME_NET*.

```
##
## Step 1: Inform Suricata about your network
##

vars:
  # more specific is better for alert accuracy and performance
  address-groups:
    HOME_NET: "192.168.1.0/24"
    #HOME_NET: "[192.168.0.0/16]"
    #HOME_NET: "[10.0.0.0/8]"
    #HOME_NET: "[172.16.0.0/12]"
    #HOME_NET: "any"
```

Figura 3.11: Configuración de Suricata

Como segundo paso, se seleccionan las salidas a habilitar. Se mantiene el directorio de registros por defecto. En cuanto a la configuración de las estadísticas, se modifica el intervalo de 8 segundos a 1 hora.

```
##
## Step 2: Select outputs to enable
##

# The default logging directory.  Any log or output file will be
# placed here if it's not specified with a full path name. This can be
# overridden with the -l command line parameter.
default-log-dir: /var/log/suricata/

# Global stats configuration
stats:
  enabled: yes
  # The interval field (in seconds) controls the interval at
  # which stats are updated in the log.
  interval: 3600
```

Figura 3.12: Configuración de Suricata (II)

Los demás campos a tener en cuenta en este paso de la configuración se revisan y no se modifican.

En el tercer paso, se configura los ajustes comunes de captura. Se establece que la interfaz es *ens33*.

```
##
## Step 3: Configure common capture settings
##
## See "Advanced Capture Options" below for more options, including Netmap
## and PF_RING.
##

# Linux high speed capture support
af-packet:
  - interface: ens33
    # Number of receive threads. "auto" uses the number of cores
    #threads: auto
    # Default clusterid. AF_PACKET will load balance packets based on flow.
    cluster-id: 99
```

Figura 3.13: Configuración de Suricata (III)

Se reinicia el servicio de Suricata con *systemctl* para que se ejecute con la nueva configuración.

3.2.2. Envío de eventos a QRadar

Para el envío de eventos a QRadar, es necesario modificar la configuración de Suricata, de forma que se habilita la salida de *eve-log* con los siguientes parámetros.

```
# Configure the type of alert (and other) logging you would like.
outputs:
  # a line based alerts log similar to Snort's fast.log
  - fast:
      enabled: yes
      filename: fast.log
      append: yes
      #filetype: regular # 'regular', 'unix_stream' or 'unix_dgram'

  # Extensible Event Format (nicknamed EVE) event log in JSON format
  - eve-log:
      enabled: yes
      filetype: syslog #regular|syslog|unix_dgram|unix_stream|redis
      filename: eve.json
      # Enable for multi-threaded eve.json output; output files are amended with
      # an identifier, e.g., eve.9.json
      #threaded: false
      #prefix: "@cee: " # prefix to prepend to each log entry
      # the following are valid when type: syslog above
      identity: "suricata"
      facility: local5
```

Figura 3.14: Configuración de Suricata (III)

Con esta configuración, se habilita la salida *eve-log* y se envían los eventos del fichero *eve.json* al *facility local5* de *syslog* (donde posteriormente, *syslog* lo envía a QRadar).

A continuación, se va a proceder a enviar la información almacenada en el *facility local5* de *syslog* a QRadar por el puerto 514. Para ello, se especifica en el fichero de configuración de *syslog* (*/etc/rsyslog.conf*), la siguiente línea *local5.* @@192.168.1.36:514*.

```
# Enviar logs a QRadar
local5.* @@192.168.1.36:514
```

Figura 3.15: Envío de logs de Suricata a QRadar

Una vez se ha modificado esta configuración, se reinician los servicios de Suricata y Syslog.

3.2.3. Creación de la fuente de datos en QRadar

En el apartado de *Log Activity* se almacenan los registros que recibe QRadar. Cuando los registros de Suricata llegan a QRadar, se encuentran sin identificar, sin nombre de evento ni fuente de datos de la que proviene.

Event Name	Log Source	Event Count	Time ▲	Low Level Category	Source IP	Source Port	Destination IP	Destination Port	Username
Unknown log event	SIM Generic Log DSM-7 :: localhost	1	Jul 16, 2023, 5:05:08 PM	Unknown Generic Log Event	192.168.1.38	0	192.168.1.38	0	N/A
Unknown log event	SIM Generic Log DSM-7 :: localhost	1	Jul 16, 2023, 5:06:54 PM	Unknown Generic Log Event	192.168.1.38	0	192.168.1.38	0	N/A
Unknown log event	SIM Generic Log DSM-7 :: localhost	1	Jul 16, 2023, 5:06:54 PM	Unknown Generic Log Event	192.168.1.38	0	192.168.1.38	0	N/A
Unknown log event	SIM Generic Log DSM-7 :: localhost	1	Jul 16, 2023, 5:08:17 PM	Unknown Generic Log Event	192.168.1.38	0	192.168.1.38	0	N/A
Unknown log event	SIM Generic Log DSM-7 :: localhost	1	Jul 16, 2023, 5:08:17 PM	Unknown Generic Log Event	192.168.1.38	0	192.168.1.38	0	N/A

Figura 3.16: Registros sin identificar

El módulo de soporte del que dispone QRadar para identificar Suricata como fuente de origen datos se encuentra disponible a partir de la versión 7.4; la que se está utilizando es la 7.3.3. Por tanto, hay que crear un tipo de fuente de datos para que QRadar identifique Suricata como fuente.

Para crearla, se accede a la pestaña *Admin*, a *QRadar Log Source Management*.

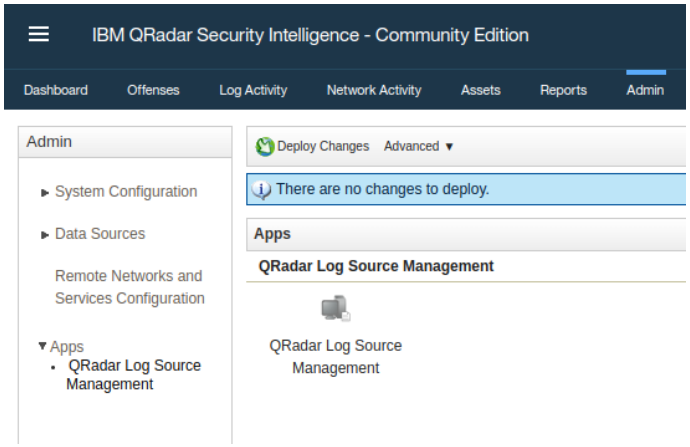


Figura 3.17: Apartado *QRadar Log Source Management*

Se crea una nueva fuente de datos con nombre *Suricata*. En el apartado *Protocol*, hay que especificar cuál es el identificador de la fuente. Se puede establecer la IP origen o nombre del equipo; en este caso, se opta por identificarlo por nombre del equipo, que es «cliente».

Figura 3.18: Creación de Suricata como fuente de datos

En *Overview*, se detallan el nombre de la fuente, el tipo de fuente de datos y el tipo de protocolo por el que recibe los eventos, los otros campos se mantienen por defecto.

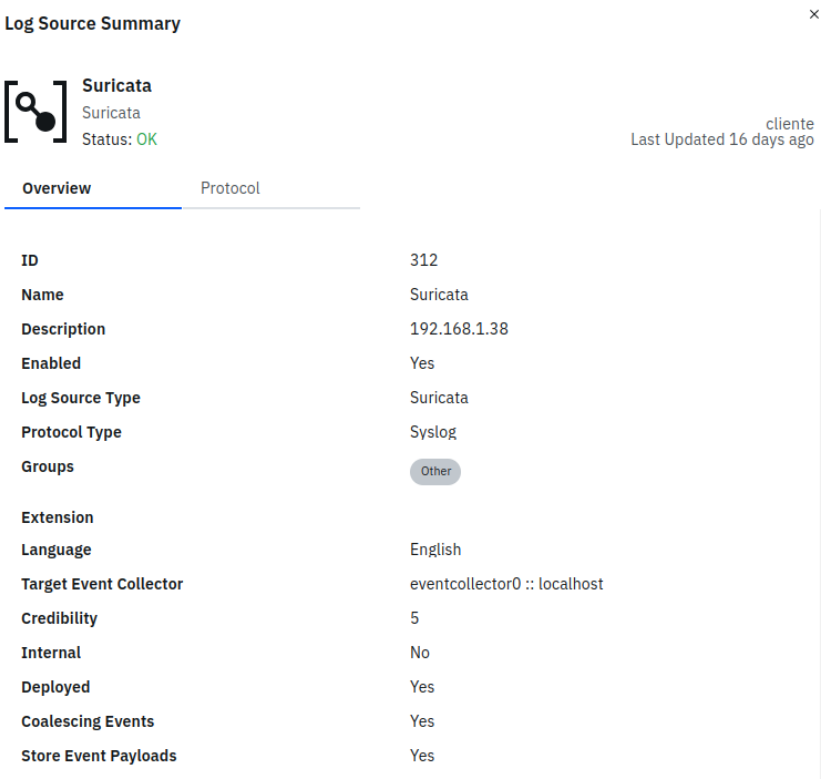


Figura 3.19: Creación de Suricata como fuente de datos (II)

Al crear una nueva fuente, se requiere de un despliegue de los cambios para que surtan efecto.

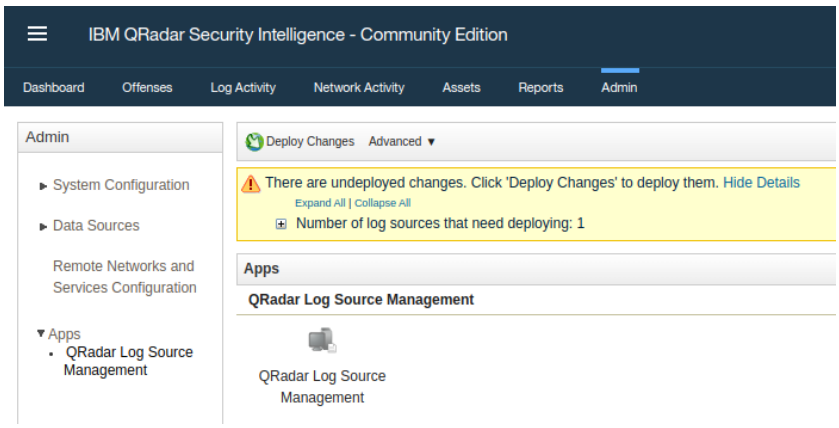


Figura 3.20: Aviso de cambios sin desplegar

Una vez se despliegan los cambios, al consultar el registro de actividad, se aprecia que ya consigue identificar la fuente de datos de Suricata.

Event Name	Log Source	Event Count	Time ▲	Low Level Category	Source IP	Source Port	Destination IP	Destination Port	Username
Unknown	Suricata	1	Jul 23, 2023, 8:48:36 AM	Unknown	192.168.1.38	0	192.168.1.38	0	N/A
Unknown	Suricata	1	Jul 23, 2023, 8:48:44 AM	Unknown	192.168.1.38	0	192.168.1.38	0	N/A
Unknown	Suricata	1	Jul 23, 2023, 8:48:52 AM	Unknown	192.168.1.38	0	192.168.1.38	0	N/A

Figura 3.21: Identificación correcta de Suricata como fuente origen

3.2.4. Mapeo de eventos en QRadar

QRadar ya identifica la fuente origen de los datos, pero no consigue distinguir los eventos, por lo que muestra en el registro el evento con nombre *Unknown*, como se aprecia en la figura 3.21. Para ello, hay que mapear los eventos. En la pestaña del registro de actividad, se seleccionan estos eventos a mapear y se abre con el *DSM Editor*.

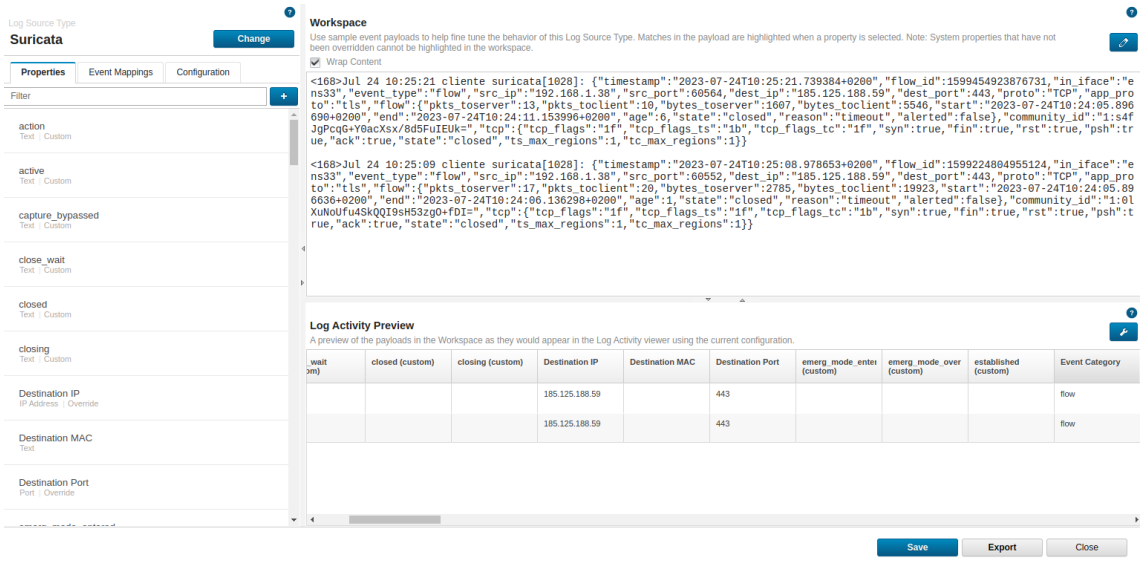


Figura 3.22: Mapeo de eventos de Suricata

Para mapear un evento, es necesario establecer los campos *Event Category* y *Event ID* con los que QRadar categorizará e identificará el evento. Estos campos los obtiene del *payload*, por tanto, hay que indicar una expresión regular para seleccionar qué parte del payload se considera *Event Category* y *Event ID*. Esta configuración se mantiene para la fuente de Suricata, no para otras fuentes.

En este caso, se ha elegido el *event_type* como *Event Category* y el protocolo (*proto*) como *Event ID*.

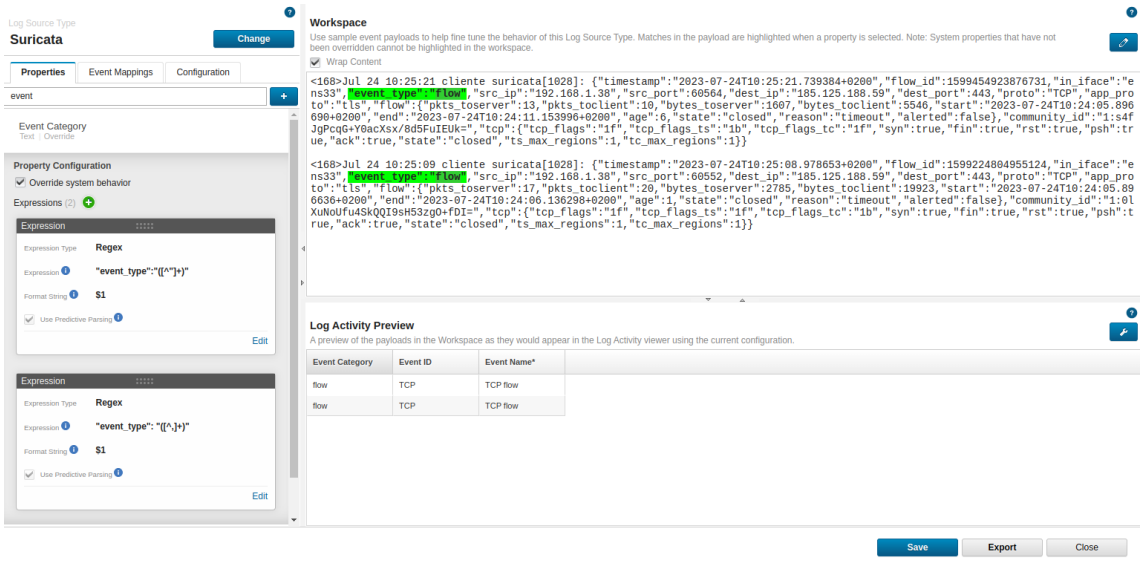


Figura 3.23: Expresiones regulares para *Event Category* - Suricata

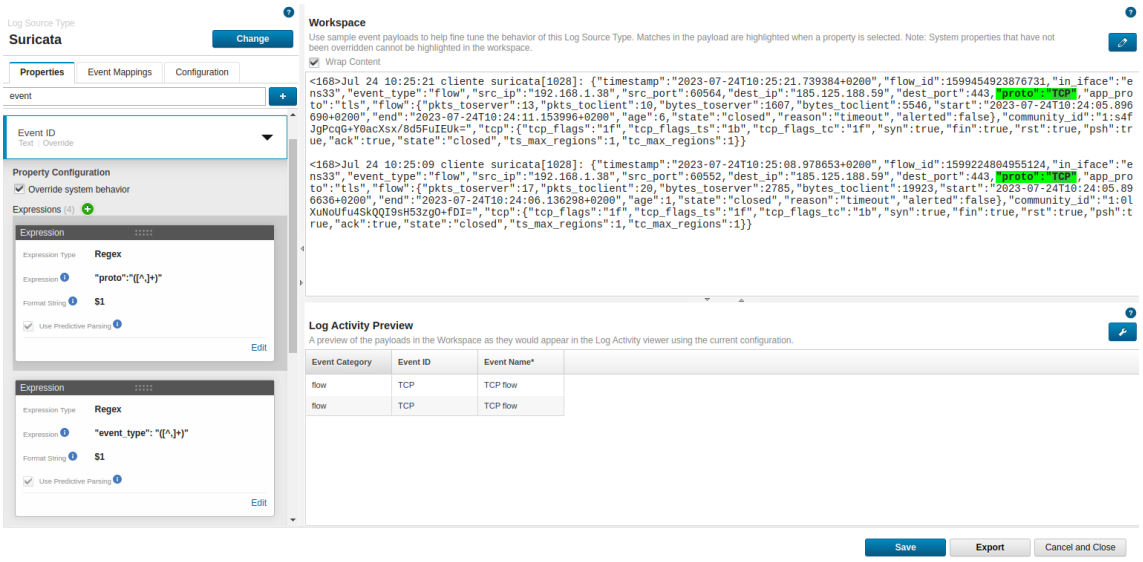


Figura 3.24: Expresiones regulares para *Event ID* - Suricata

Dependiendo del *payload*, el tipo de evento y el protocolo pueden aparecer de diferentes formas. Por ello, para cada configuración, se añaden múltiples expresiones regulares que identifiquen estos campos independientemente del tipo de evento de Suricata que aparezca. Se añaden todas las expresiones regulares y se ordenan, de forma que si no encuentra el campo con la primera expresión, prueba con la siguiente hasta encontrar la que coincida.

Con la categoría e identificación del registro, ya se puede crear el evento. Hay que indicar el valor del *Event ID* y el valor del *Event Category* y se asocia a un registro QID.

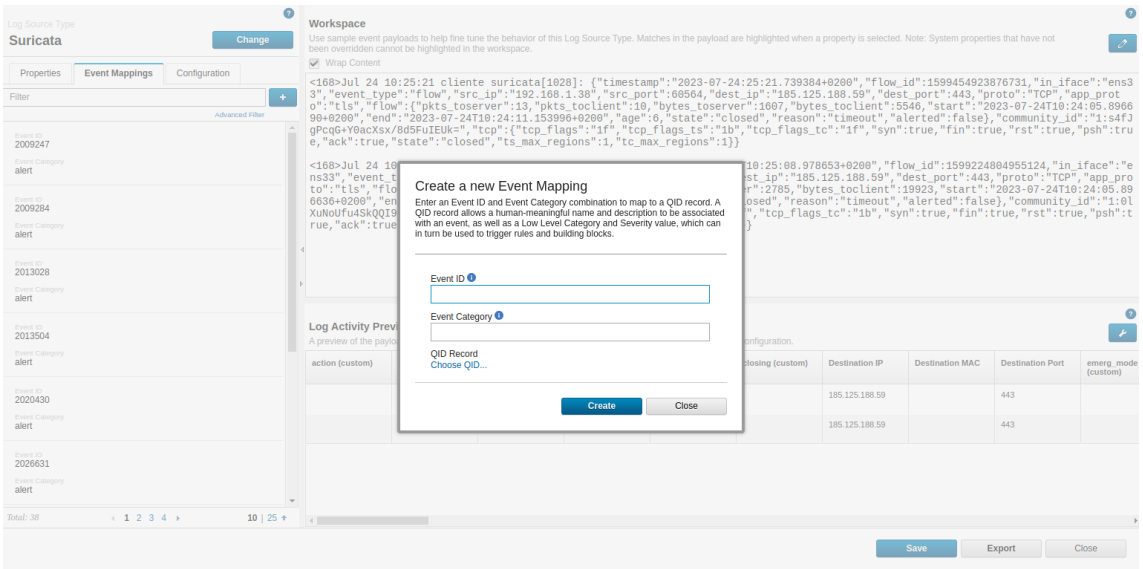


Figura 3.25: Creación de nuevo mapeo de evento - Suricata

Hay veces que no se encuentra el QID que se necesita, por lo que hay que crearlo. Se requiere el nombre, la fuente de datos a la que va asociada, los niveles de categoría y la severidad.

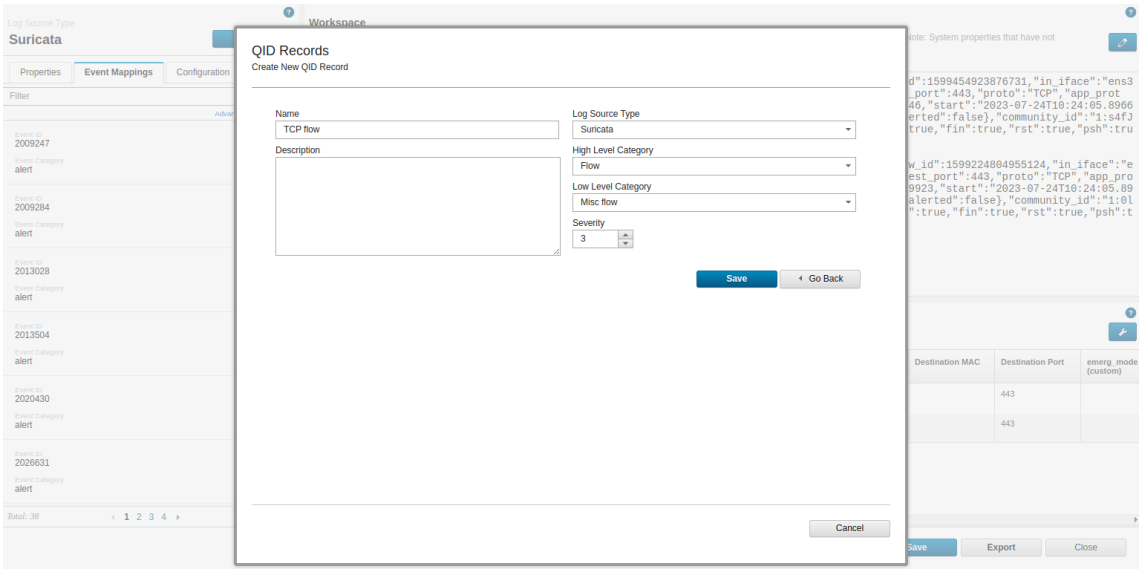


Figura 3.26: Creación de QID - Suricata

Una vez creado, se selecciona como QID.

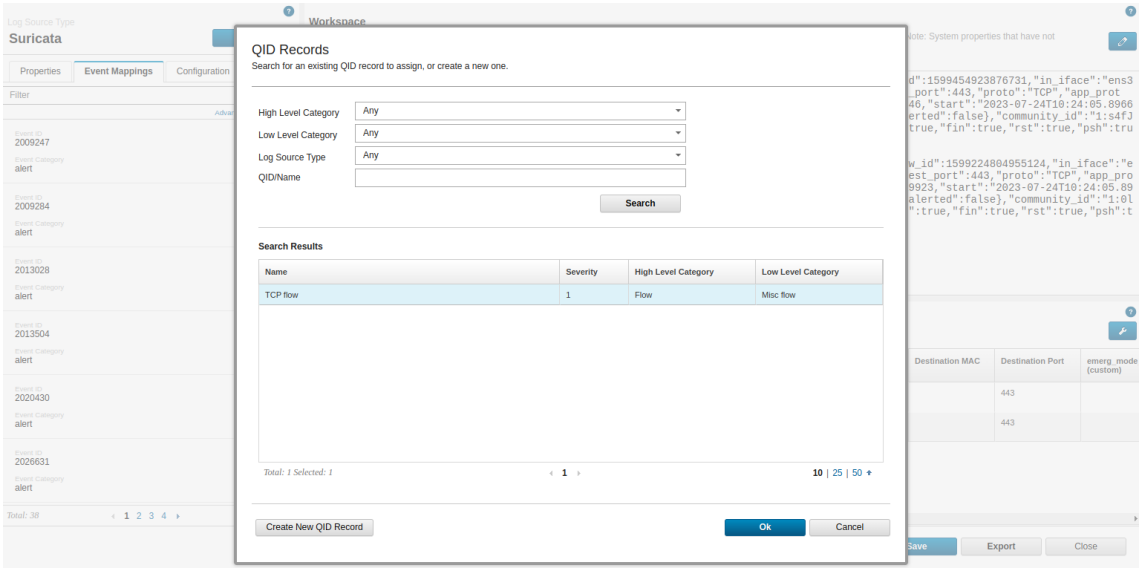


Figura 3.27: Selección de QID - Suricata

Ya se tienen todos los campos esenciales para el mapeo del evento.

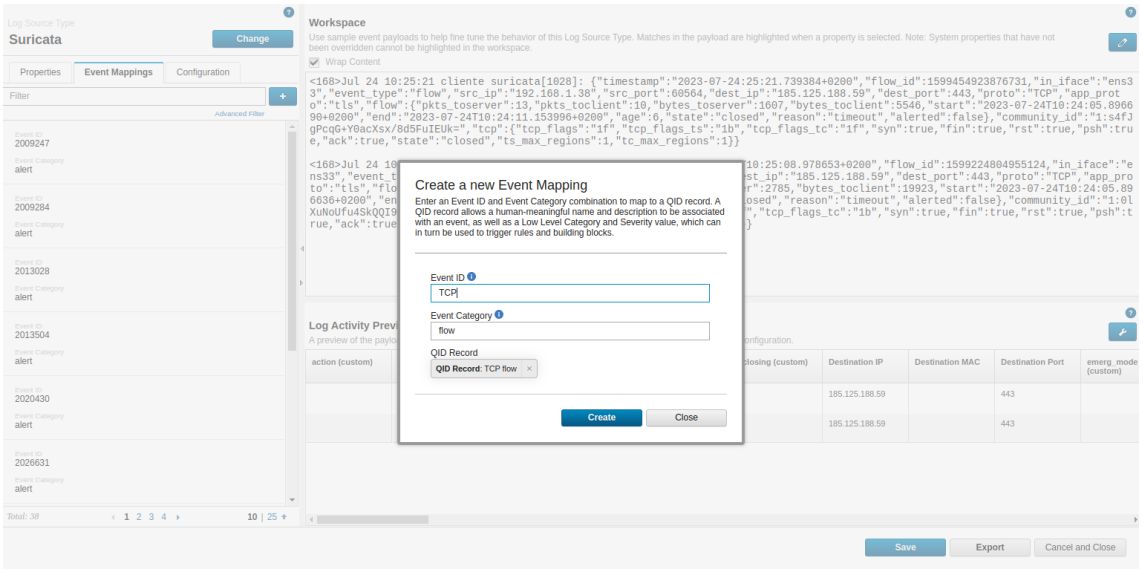


Figura 3.28: Creación de nuevo mapeo (II) - Suricata

Para los eventos que se tratan de alertas, se escoge un identificador diferente. En los eventos de alertas, aparece el identificador de la regla, por lo tanto, se decide establecer este número como *Event ID* de forma que todas las alertas que se generen con una misma regla, tengan el mismo nombre y categoría.

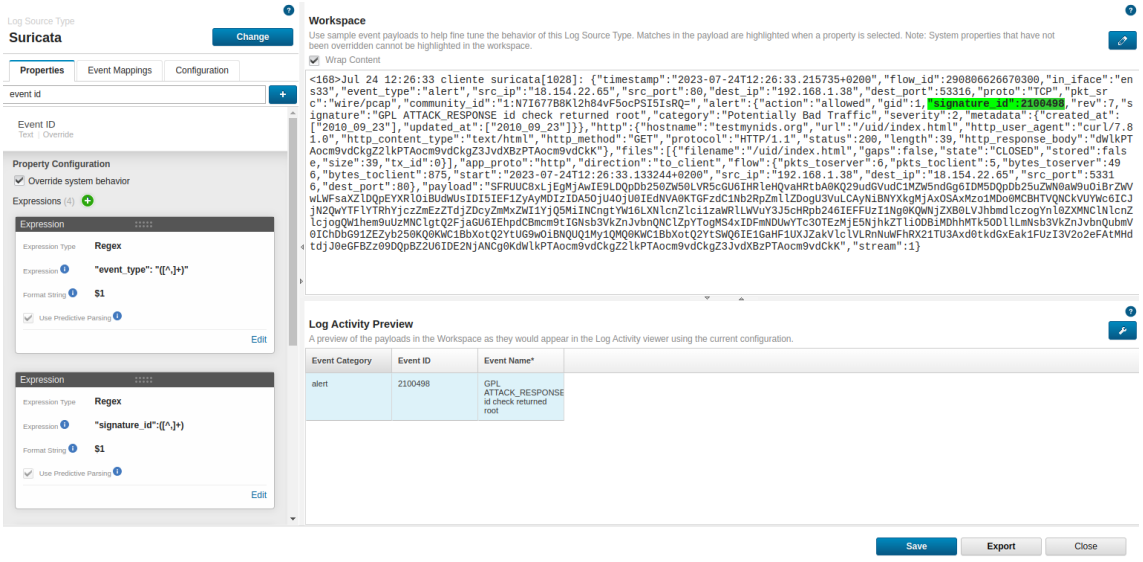


Figura 3.29: Mapeo de alerta de Suricata

Estos pasos para mapear se realiza con todos los eventos que se generan, quedando el registro de actividades como se muestra en la siguiente figura.

Event Name	Log Source	Event Count	Time	Low Level Category	Source IP	Source Port	Destination IP	Destination Port	Username
TLS communication	Suricata	1	Aug 30, 2023, 6:26:46 PM	Misc Network Communication Event	192.168.1.38	39622	3.220.77.232	443	N/A
TLS communication	Suricata	1	Aug 30, 2023, 6:26:46 PM	Misc Network Communication Event	192.168.1.38	39622	34.120.115.102	443	N/A
TLS communication	Suricata	1	Aug 30, 2023, 6:26:46 PM	Misc Network Communication Event	192.168.1.38	39622	34.120.208.123	443	N/A
TLS communication	Suricata	1	Aug 30, 2023, 6:26:46 PM	Misc Network Communication Event	192.168.1.38	39622	142.250.184....	443	N/A
TLS communication	Suricata	1	Aug 30, 2023, 6:26:46 PM	Misc Network Communication Event	192.168.1.38	39622	34.117.237.239	443	N/A
DNS connection	Suricata	24	Aug 30, 2023, 6:26:46 PM	Misc Network Communication Event	192.168.1.38	39622	80.58.61.250	53	N/A
Suricata stats	Suricata	1	Aug 30, 2023, 6:26:43 PM	System Status	192.168.1.38	0	192.168.1.38	0	N/A
Suricata stats	Suricata	2	Aug 30, 2023, 6:26:27 PM	System Status	192.168.1.38	0	192.168.1.38	0	N/A
Suricata stats	Suricata	2	Aug 30, 2023, 6:26:11 PM	System Status	192.168.1.38	0	192.168.1.38	0	N/A
UDP flow	Suricata	1	Aug 30, 2023, 6:26:04 PM	Misc flow	192.168.1.38	39622	80.58.61.250	53	N/A
DNS connection	Suricata	2	Aug 30, 2023, 6:26:03 PM	Misc Network Communication Event	192.168.1.38	39622	80.58.61.250	53	N/A

Figura 3.30: Registro de actividad con eventos de Suricata

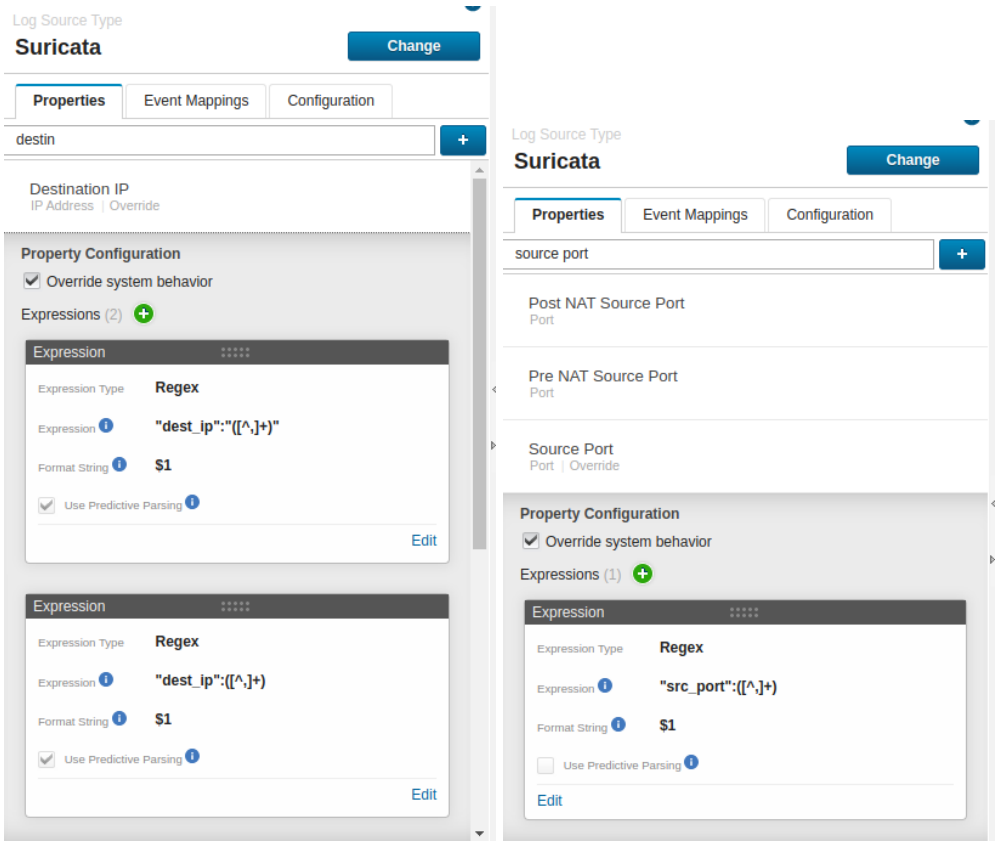
3.2.5. Parseo de campos en QRadar

Hay parámetros que aparecen en el *payload* que no se encuentran como propiedades del evento o existen estas propiedades pero no se detecta bien. Para ello, se realiza el parseo de campos mediante expresiones regulares.

Algunos de los campos que ya existían como propiedad pero no se detectaban bien y han requerido de un parseo son:

- *Source IP* y *Source Port*
- *Destination IP* y *Destination Port*
- *Protocol*

Las expresiones regulares para el parseo de algunos de estos campos se realiza como se muestra a continuación:



(a) *Destination IP* **(b)** *Source Port*

Figura 3.31: Parseo de campos de Suricata

Otros campos interesantes que no están como propiedad y se crean son: *action*, *user-agent* y *url*. Para crearlos, se necesita un nombre de propiedad y el tipo del campo, además se habilita la propiedad para que se pueda usar en reglas y para indexar en búsquedas.

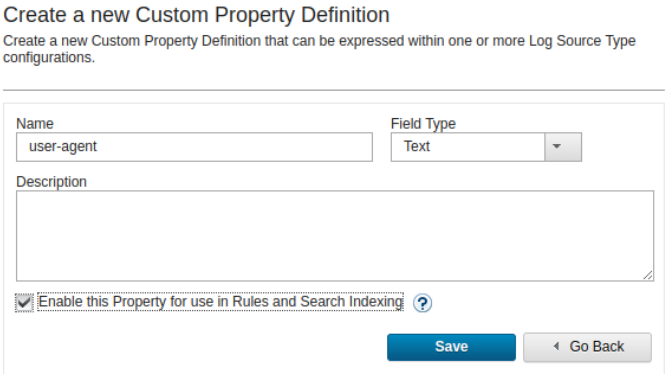
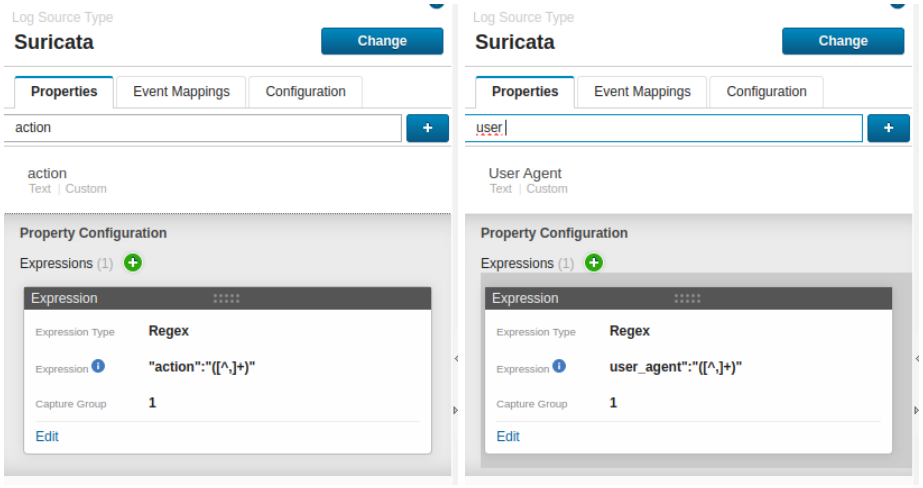
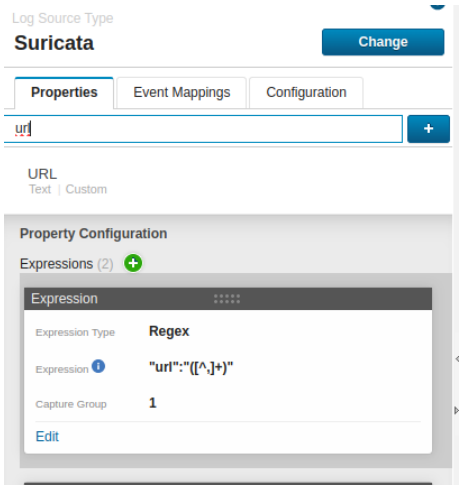


Figura 3.32: Creación de propiedad en Suricata

Se parsea los campos con las siguientes expresiones regulares:



(a) *action* **(b)** *user-agent*



(c) *url*

Figura 3.33: Parseo de campos de Suricata (II)

En la siguiente figura, se muestran cómo quedan los campos parseados:

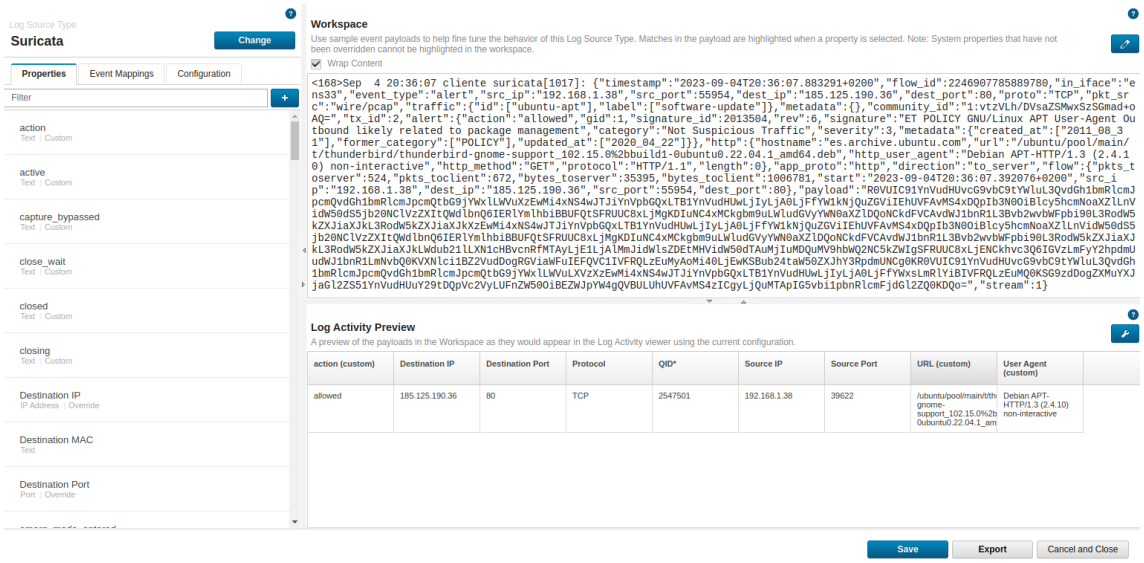


Figura 3.34: Campos parseados

3.2.6. Creación de reglas en QRadar

La creación de reglas se efectúa en la sección de *Offenses*, en el apartado de *Rules*.

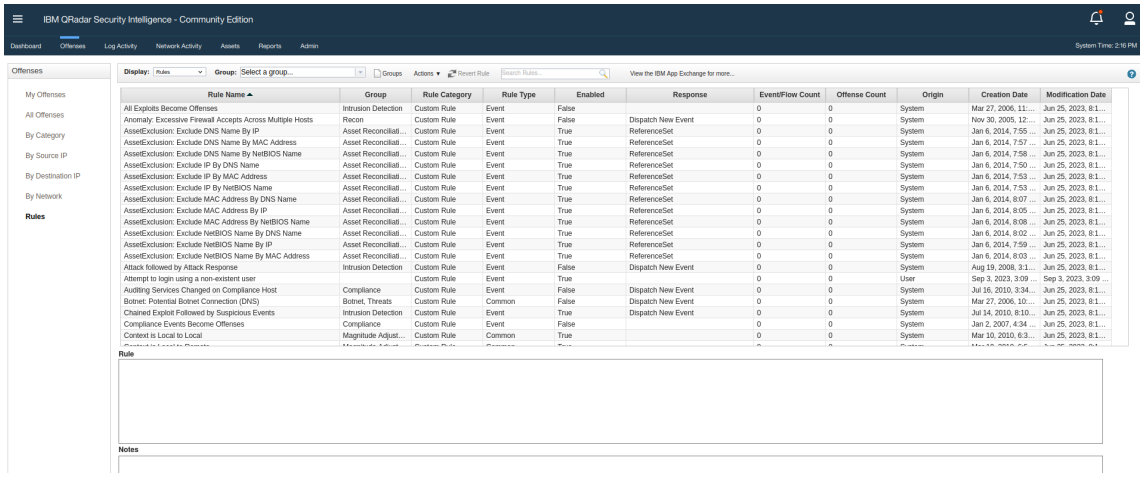


Figura 3.35: Pestaña *Offenses*, apartado *Rules*

Se crea una nueva regla a partir de eventos. Primero se establece el nombre de la regla, la lógica y qué condiciones se tienen que cumplir para generar la regla. Como algunos de los eventos ya son alertas generadas por reglas en Suricata, solo es necesario añadir el nombre de la regla, el contexto y el QID asociado al evento.

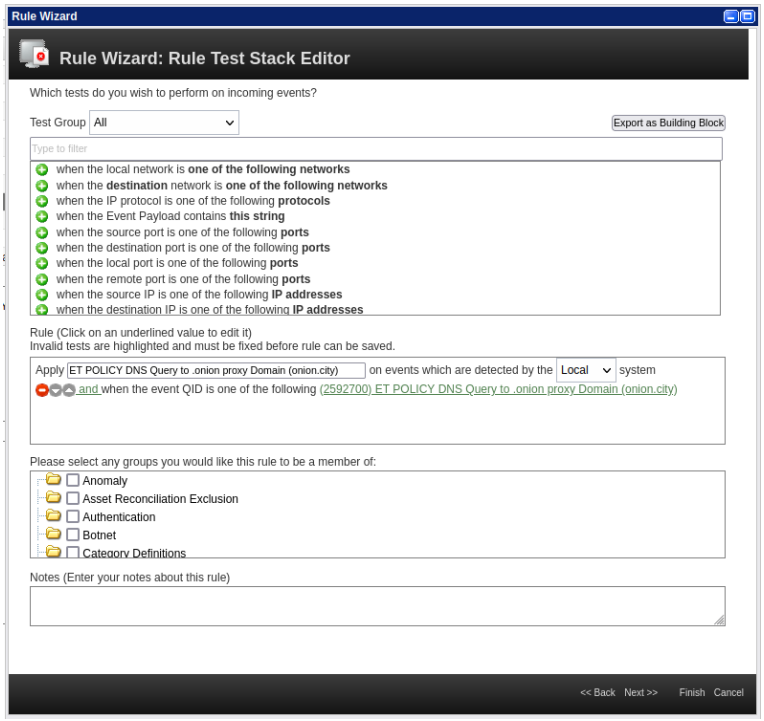


Figura 3.36: Regla *ET POLICY DNS Query to .onion proxy Domain (onion.city)*

El segundo paso consiste en añadir una severidad, credibilidad y relevancia y generar una ofensa cuando se cumpla la regla. Esta ofensa estará indexada por IP origen.

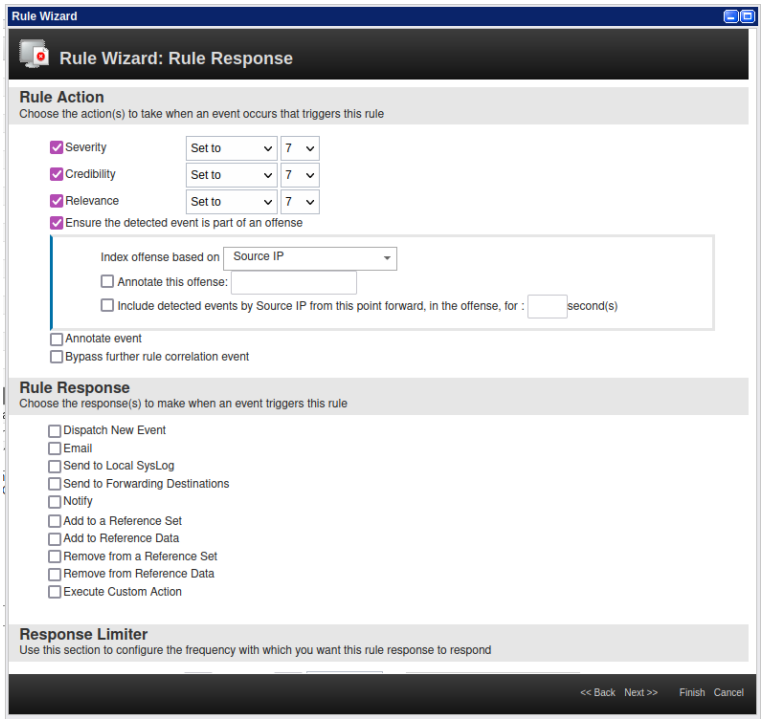


Figura 3.37: Regla *ET POLICY DNS Query to .onion proxy Domain (onion.city)* (II)

Con estos pasos, se crea la regla *GET POLICY DNS Query to .onion proxy Domain (onion.city)*, para detectar peticiones DNS en las que se resuelva el dominio *onion.city*.

Se elaboran múltiples reglas, entre las que están *GLP ATTACK RESPONSE id check returned root* y *SSH on unusual port*.

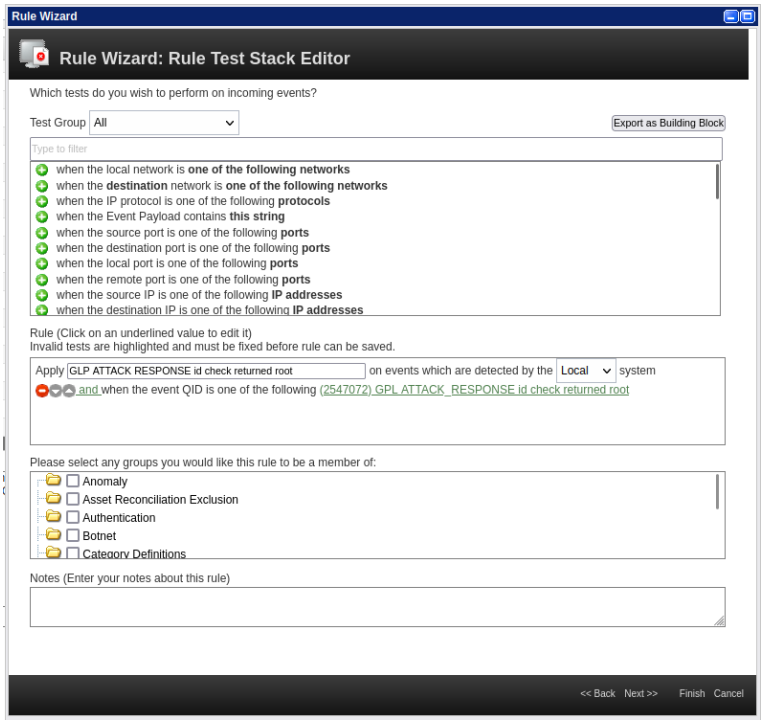


Figura 3.38: Regla *GLP ATTACK RESPONSE id check returned root*

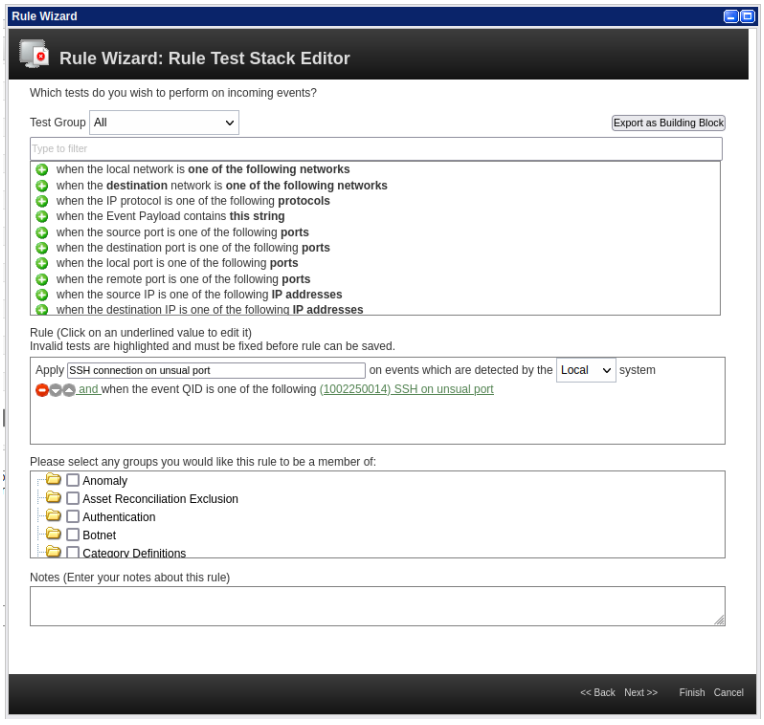


Figura 3.39: Regla *SSH on unusual port*

3.3 OSSEC

Se utiliza OSSEC en el modo *cliente-servidor*. Por ello, el servidor de OSSEC se instala en una máquina Ubuntu 22.04, que actúa como centralita y recoge toda la información del agente en la máquina del cliente para luego transmitirlo a QRadar, mientras que el agente OSSEC se instala en la máquina del cliente, junto a Suricata.

Se realiza la instalación de OSSEC tal y como se muestra en el apartado de OSSEC de la sección 2.2.2. La única diferencia es que, a la hora de elegir el tipo de instalación, se seleccionará *servidor* para el servidor OSSEC, y *agente* para el agente de OSSEC que se instala en la máquina del cliente.

3.3.1. Gestión de comunicación servidor-agente

La comunicación entre los agentes y el servidor OSSEC se produce generalmente en el puerto 1514/udp en modo seguro [56].

Para conectar agente y servidor, hay que seguir los siguiente pasos [57]:

1. Ejecute *manage_agents* en el servidor OSSEC.
2. Agregar un agente.
3. Extraer la llave del agente.
4. Copiar esa clave en el agente.
5. Ejecutar *manage_agents* en el agente.
6. Importar la clave copiada del gerente.
7. Reiniciar los procesos del servidor OSSEC.
8. Iniciar el agente.

Al hacer el primer paso, *manage_agents* muestra una interfaz que solicita el identificador del agente, el nombre y la IP.

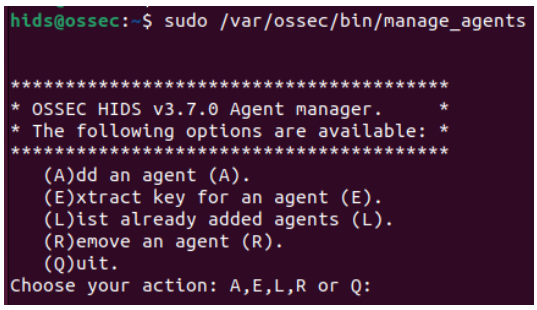


Figura 3.40: Añadir agente al servidor OSSEC

Una vez añadido, se puede consultar los agentes disponibles:

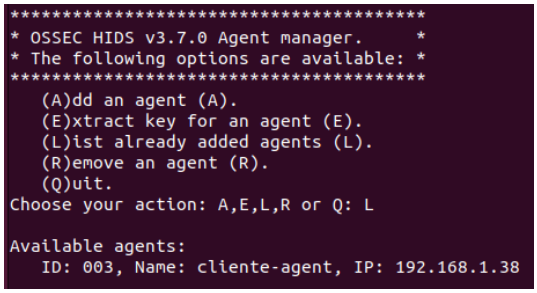


Figura 3.41: Listar agentes del servidor OSSEC

Se reinicia el servidor con */var/ossec/bin/ossec-control restart*.

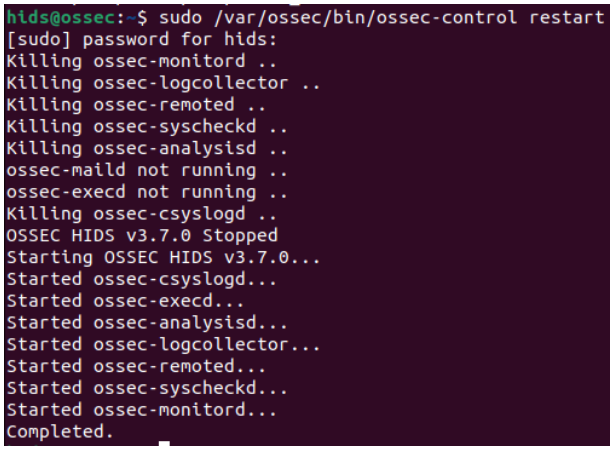


Figura 3.42: Reinicio del servidor OSSEC

En la máquina cliente con el agente de OSSEC, se utiliza el mismo archivo *manage_agents* para importar la clave que se ha generado al crear el agente en el servidor.

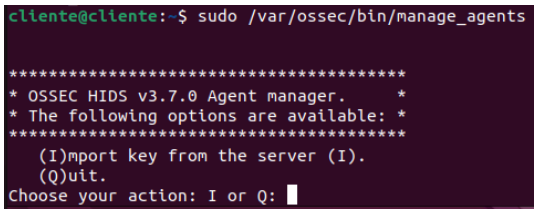


Figura 3.43: Importar clave en el agente OSSEC

Se observa que, aunque se trata del mismo fichero en la misma ubicación, las opciones de las que se dispone en el agente de OSSEC son diferentes a las del servidor.

3.3.2. Envío de alertas a QRadar

El envío de alertas a QRadar solo se puede realizar desde un servidor OSSEC o con una instalación en modo local y se realiza mediante la habilitación de la salida *syslog* en el fichero de configuración [58].

Se especifica la IP de QRadar y el puerto 514, el formato de la salida en *JSON* y se establece que se envíen las alertas que tengan un nivel 3 o más.

```
<syslog_output>
      <server>192.168.1.36</server>
      <port>514</port>
      <level>3</level>
      <format>json</format>
</syslog_output>
```

Figura 3.44: Envío de alertas de OSSEC a QRadar

3.3.3. Creación de la fuente de datos en QRadar

Cuando las alertas de OSSEC llegan a QRadar, se encuentran sin identificar, sin nombre de evento ni fuente de datos de la que proviene.

Event Name	Log Source	Event Count	Time ▲	Low Level Category	Source IP	Source Port	Destination IP	Destination Port	Username
Unknown log event	SIM Generic Log DSM-7 :: localhost	1	Jul 25, 2023, 4:35:51 PM	Unknown Generic Log Event	192.168.1.42	0	192.168.1.42	0	N/A
Unknown log event	SIM Generic Log DSM-7 :: localhost	1	Jul 25, 2023, 4:35:27 PM	Unknown Generic Log Event	192.168.1.42	0	192.168.1.42	0	N/A
Unknown log event	SIM Generic Log DSM-7 :: localhost	1	Jul 25, 2023, 4:35:27 PM	Unknown Generic Log Event	192.168.1.42	0	192.168.1.42	0	N/A

Figura 3.45: Registros sin identificar (II)

Aunque QRadar disponga de un módulo para OSSEC, parece que en este caso no logra identificar los eventos como de OSSEC. Al igual que con Suricata, hay que crear la fuente origen de datos.

Se crea una nueva fuente de datos con nombre *OSSEC*. En el apartado *Protocol*, se ha optado por identificar la fuente por nombre del equipo, que es «ossec».

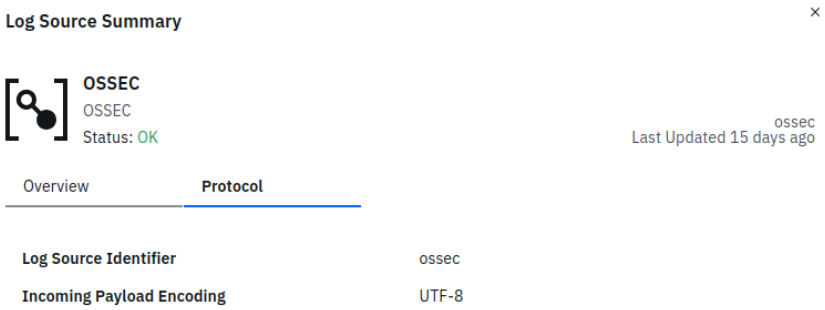


Figura 3.46: Creación de OSSEC como fuente de datos

En *Overview*, se detallan el nombre de la fuente, el tipo de fuente de datos y el tipo de protocolo por el que recibe los eventos, los otros campos se han mantenido por defecto.

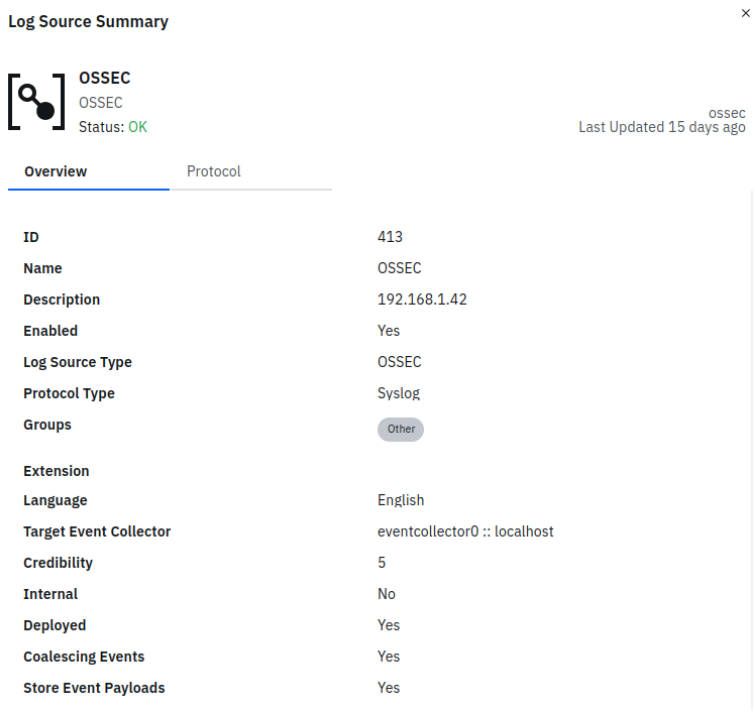


Figura 3.47: Creación de OSSEC como fuente de datos (II)

Se realiza el despliegue de los cambios y ya se identifica OSSEC como fuente origen de datos.

Event Name	Log Source	Event Count	Time ▲	Low Level Category	Source IP	Source Port	Destination IP	Destination Port	Username
Unknown	OSSEC	1	Jul 28, 2023, 10:06:45 AM	Unknown	192.168.1.42	0	192.168.1.42	0	N/A
Unknown	OSSEC	1	Jul 28, 2023, 10:07:23 AM	Unknown	192.168.1.42	0	192.168.1.42	0	N/A
Unknown	OSSEC	1	Jul 28, 2023, 10:07:57 AM	Unknown	192.168.1.42	0	192.168.1.42	0	N/A

Figura 3.48: Identificación correcta de OSSEC como fuente origen

3.3.4. Mapeo de eventos en QRadar

Aunque reconozca la fuente de OSSEC, desconoce la información que hay en los registros. Para el mapeo de eventos de OSSEC, se sigue el mismo procedimiento que para el mapeo en Suricata.

Como *Event Category*, se escoge el nombre del equipo y la fuente, que es *ossec*.

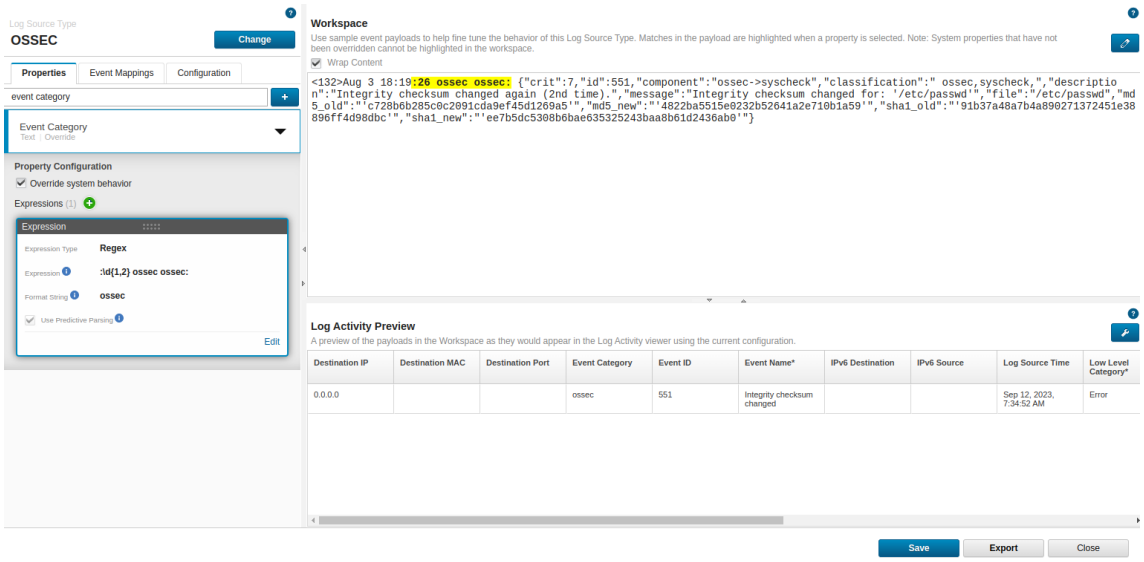


Figura 3.49: Expresiones regulares para *Event Category* - OSSEC

OSSEC lo que envía a QRadar son las alertas que detecta y que tienen un nivel mayor que 3. Estas alertas tienen asignado un identificador, que hace referencia a la regla que la genera, por tanto, se selecciona este campo *id* como *Event ID*.

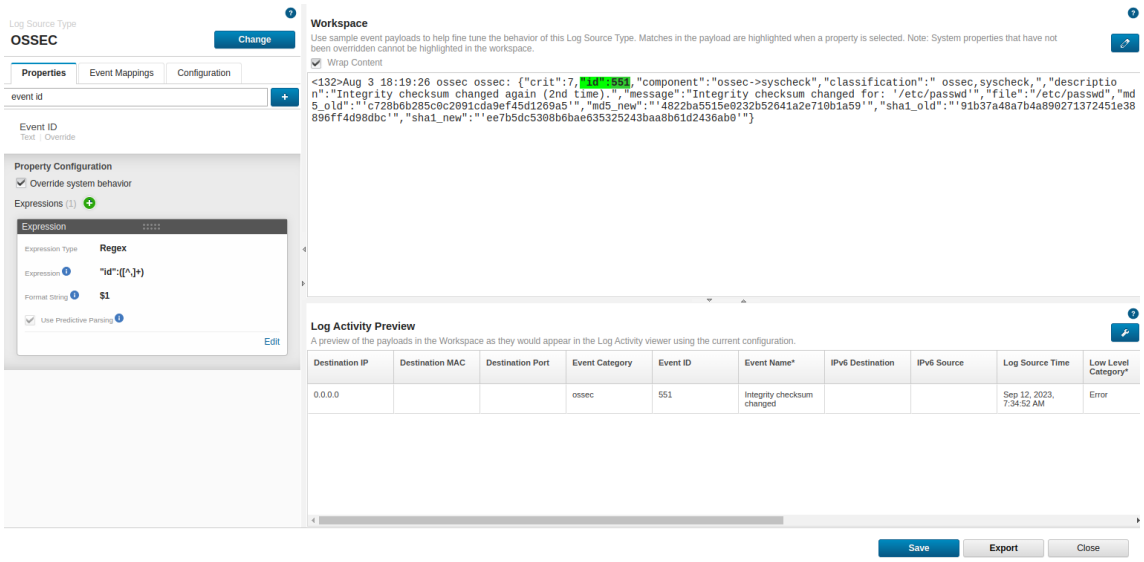


Figura 3.50: Expresiones regulares para *Event ID* - OSSEC

Para la selección de QID, al tener OSSEC un módulo de soporte, no es necesaria la generación de un QID específico, con seleccionar la fuente de OSSEC y buscar por el nombre de la regla, se encuentra el QID con el que asociar el evento.

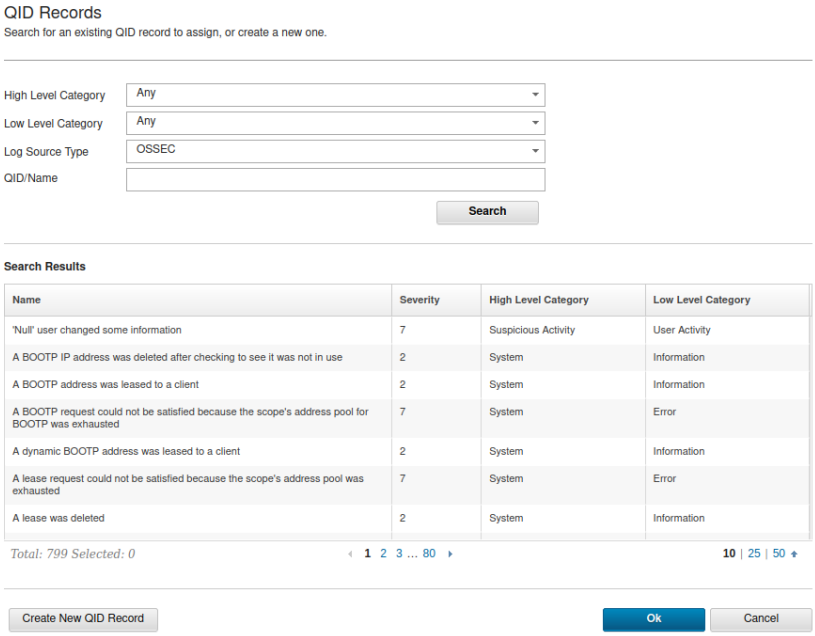


Figura 3.51: Registros QID de OSSEC

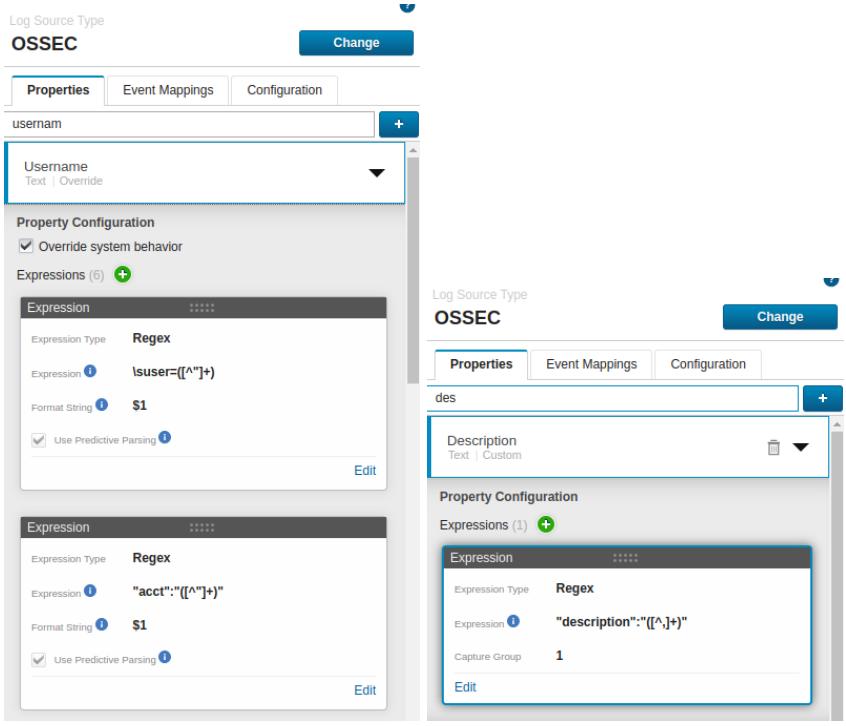
Para mapear un evento no es necesario tener el *payload*, conociendo el identificador de la regla es suficiente para mapearlo. Estos pasos se realizan con algunas de las alertas de OSSEC, quedando el registro de actividades como se muestra en la siguiente figura.

Event Name	Log Source	Event Count	Time ▲	Low Level Category	Source IP	Source Port	Destination IP	Destination Port	Username
Ossec server started	OSSEC	1	Sep 2, 2023, 8:53:54 AM	Service Started	192.168.1.42	0	192.168.1.42	0	N/A
Login session opened	OSSEC	1	Sep 2, 2023, 8:53:58 AM	Session Opened	192.168.1.42	0	192.168.1.42	0	N/A
User authentication failure	OSSEC	1	Sep 2, 2023, 9:03:19 AM	User Login Failure	192.168.1.42	0	192.168.1.42	0	N/A
User authentication failure	OSSEC	1	Sep 2, 2023, 9:03:34 AM	User Login Failure	192.168.1.42	0	192.168.1.42	0	N/A
User authentication failure	OSSEC	1	Sep 2, 2023, 9:03:39 AM	User Login Failure	192.168.1.42	0	192.168.1.42	0	N/A
User authentication failure	OSSEC	3	Sep 2, 2023, 9:03:44 AM	User Login Failure	192.168.1.42	0	192.168.1.42	0	N/A
User login failed	OSSEC	1	Sep 2, 2023, 9:04:44 AM	User Login Failure	192.168.1.37	0	192.168.1.42	0	N/A
User missed the password more than one time	OSSEC	1	Sep 2, 2023, 9:04:59 AM	User Login Failure	192.168.1.37	0	192.168.1.42	0	N/A
User authentication failure	OSSEC	1	Sep 2, 2023, 9:44:47 AM	User Login Failure	192.168.1.42	0	192.168.1.42	0	N/A
User authentication failure	OSSEC	1	Sep 2, 2023, 9:45:03 AM	User Login Failure	192.168.1.42	0	192.168.1.42	0	N/A
User authentication failure	OSSEC	1	Sep 2, 2023, 11:00:46 AM	User Login Failure	192.168.1.42	0	192.168.1.42	0	N/A
Ossec server started	OSSEC	1	Sep 3, 2023, 8:40:12 AM	Service Started	192.168.1.42	0	192.168.1.42	0	N/A
Ossec agent started	OSSEC	1	Sep 3, 2023, 10:08:04 AM	Information	192.168.1.42	0	192.168.1.42	0	N/A
Login session opened	OSSEC	1	Sep 3, 2023, 10:08:08 AM	Session Opened	192.168.1.42	0	192.168.1.42	0	N/A
Apparmor DENIED	OSSEC	1	Sep 3, 2023, 10:08:12 AM	File Access Failure	192.168.1.42	0	192.168.1.42	0	N/A
Apparmor DENIED	OSSEC	1	Sep 3, 2023, 10:08:12 AM	File Access Failure	192.168.1.42	0	192.168.1.42	0	N/A

Figura 3.52: Registro de actividad con eventos de OSSEC

3.3.5. Parseo de campos en QRadar

En OSSEC, al igual que en Suricata, se realiza la creación y el parseo de algunas de las propiedades. Algunas de estas propiedades que se han creado y/o parseado son *username* y *description*, quedando como se muestra a continuación.



(a) *username* **(b)** *description*

Figura 3.53: Parseo de campos de OSSEC

3.3.6. Creación de reglas en QRadar

Una de las reglas que se crea es *Succesful sudo to ROOT executed after failed login attempts*, que identifica una operación satisfactoria desde un usuario *root* tras haberse registrado intentos de acceso fallidos en menos de 5 minutos. Para esta regla, se tiene en cuenta otras reglas como *Succesfull sudo to ROOT executed* y *Login failed*, entre otras.

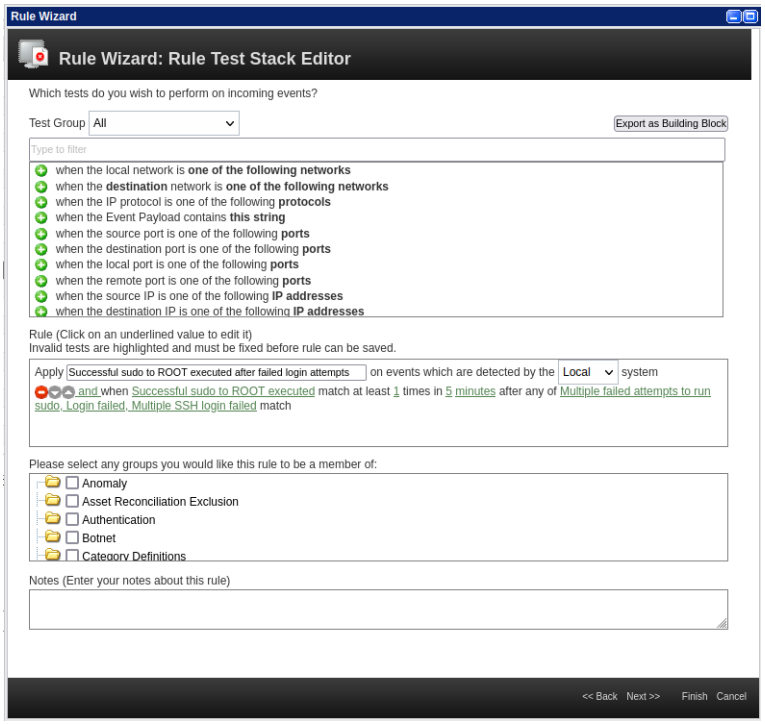


Figura 3.54: Regla *Succesful sudo to ROOT executed after failed login attempts*

Otras de las reglas que se han creado consisten en identificar modificaciones en el *crontab* y accesos fallidos desde usuarios no existentes.

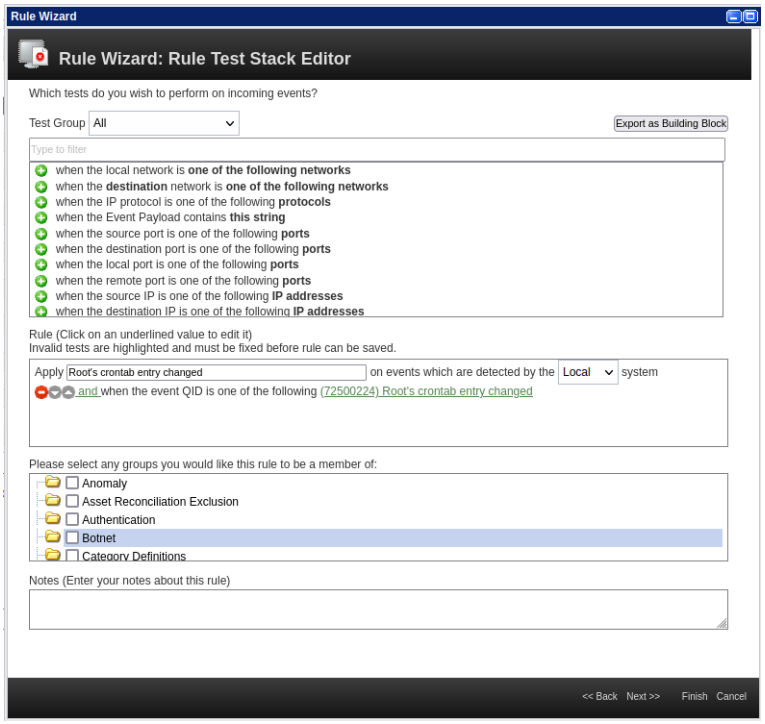


Figura 3.55: Regla *Root's crontab entry changed*

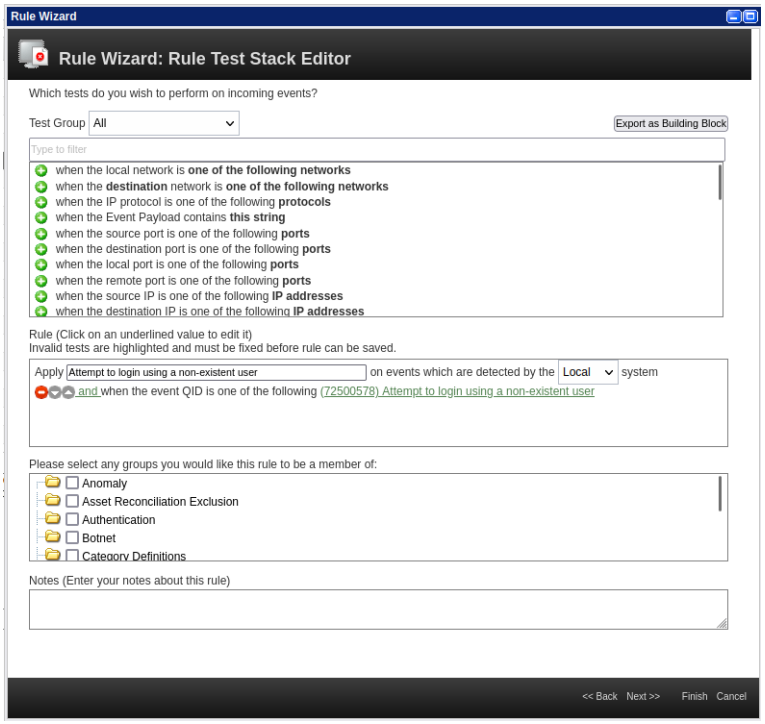


Figura 3.56: Regla *Attempt to login using a non-existen user*

CAPÍTULO 4
Pruebas

En este capítulo se simulan actividades y comportamientos sospechosos para comprobar que:

- las herramientas Suricata y OSSEC son capaces de detectar el comportamiento que se está simulando;
- los eventos se registran en QRadar correctamente;
- la regla genera una ofensa correctamente.

Para realizar el testeo de este proyecto, se realizará una serie de pruebas en la máquina *cliente*. Se comienza probando con Suricata para proseguir con OSSEC.

4.1 Suricata

4.1.1. Regla *ET POLICY CURL User Agent*

Para probar esta regla, en la máquina *cliente* se ejecuta *curl http://testmynids.org/uid/index*. Esta web genera la respuesta *uid=0(root) gid=0(root) groups=0(root)*.

```
cliente@cliente:~/Desktop$ curl http://testmynids.org/uid/index
<!doctype html>
<html>
    <head>
        <link rel="stylesheet" type="text/css" href="main.css" />
        <title>testmyNIDS.org | tmNIDS.sh</title>
    </head>

    <body>
        <article>
            <h1>Hey! What's up?</h1>
            <div>
                <p>This page is just a placeholder, as this website doesn't have
 anything worth browsing. Its purpose is explained in the project <a href="https
://github.com/3CORESec/testmynids.org">Github page</a>. We do <strong>not</stron
g> host any illegal or malicious content.</p>
                <p>&mdash; <a href="https://twitter.com/3CORESec">@3CORESec</a><
/p>
            </div>
        </article>
    </body>
</html>
```

Figura 4.1: Simulación con *curl*

En el registro de actividades, aparece el evento.

Event Name	Log Source	Event Count	Time	Low Level Category	Source IP	Source Port	Destination IP	Destination Port
ET POLICY CURL User Agent	Suricata	1	Sep 12, 2023, 5:55:58 PM	Misc Recon Event	192.168.1.38	56402	18.154.22.65	80

Figura 4.2: Evento *ET POLICY CURL User Agent*

Se genera la ofensa:

All Offenses > Offense 18 (Summary)

Offense 18 — Summary | Display | Events | Flows | Actions | Print

Magnitude		Status		Relevance	5	Severity	9	Credibility	3
Description	ET POLICY CURL User Agent	Offense Type	Destination IP						
		Event/Flow count	1 events and 0 flows in 1 categories						
Source IP(s)	192.168.1.38	Start	Sep 12, 2023, 5:55:58 PM						
Destination IP(s)	18.154.22.65	Duration	0s						
Network(s)	other	Assigned to	Unassigned						

Offense Source Summary

IP	Offenses
18.154.22.65	1

Figura 4.3: Ofensa *ET POLICY CURL User Agent*

4.1.2. Regla *SSH connection on unsual port*

La regla detecta una conexión SSH desde un puerto inusual. Para esto, se cabmia el puerto de SSH al 2223. Se establece la conexión desde el host a la máquina virtual del cliente.

```
paula@autumn:~$ ssh cliente@192.168.1.38 -p 2223
cliente@192.168.1.38's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-32-generic x86_64)

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

Expanded Security Maintenance for Applications is not enabled.

3 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Sun Sep  3 16:43:45 2023 from 192.168.1.37
cliente@cliente:~$
```

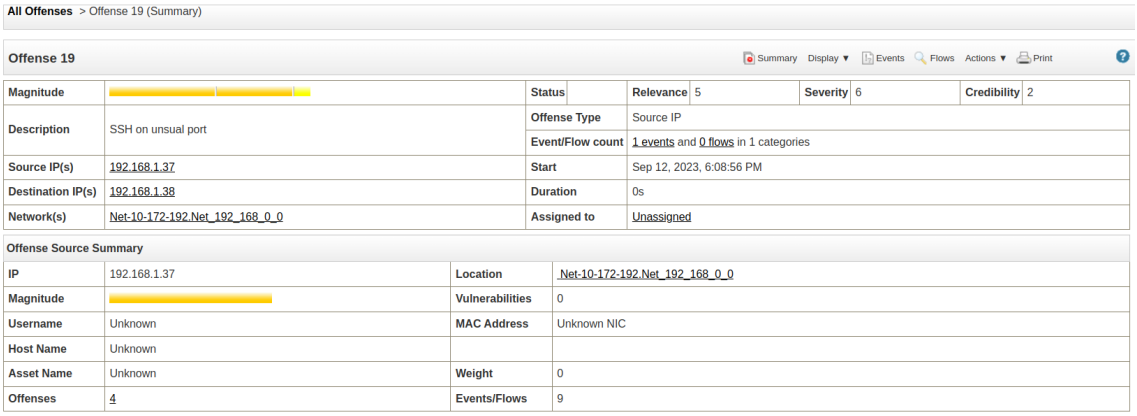
Figura 4.4: Conexión SSH por el puerto 2223

En el registro de actividades, aparece el evento.

Event Name	Log Source	Event Count	Time	Low Level Category	Source IP	Source Port	Destination IP	Destination Port
SSH connection	Suricata	1	Sep 12, 2023, 6:08:56 PM	Session Opened	192.168.1.37	36086	192.168.1.38	2223
SSH on unsual port	Suricata	1	Sep 12, 2023, 6:08:56 PM	Unauthorized Access Attempt	192.168.1.37	36086	192.168.1.38	2223

Figura 4.5: Evento *SSH on unsual port*

Se genera la ofensa:



All Offenses > Offense 19 (Summary)

Offense 19

Summary | Display | Events | Flows | Actions | Print

Magnitude	[redacted]	**Status**		**Relevance** 5	**Severity** 6	**Credibility** 2	
Description	SSH on unsual port	**Offense Type**	Source IP				
		Event/Flow count	1 events and 0 flows in 1 categories				
Source IP(s)	192.168.1.37	**Start**	Sep 12, 2023, 6:08:56 PM				
Destination IP(s)	192.168.1.38	**Duration**	0s				
Network(s)	Net-10-172-192.Net_192_168_0_0	**Assigned to**	Unassigned				

Offense Source Summary

IP	192.168.1.37	**Location**	Net-10-172-192.Net_192_168_0_0
Magnitude	[redacted]	**Vulnerabilities**	0
Username	Unknown	**MAC Address**	Unknown NIC
Host Name	Unknown		
Asset Name	Unknown	**Weight**	0
Offenses	4	**Events/Flows**	9

Figura 4.6: Ofensa *SSH on unsual port*

4.1.3. Regla *ET POLICY DNS Query to .onion proxy Domain (onion.city)*

Con esta regla, se detectan accesos a dominios *onion.city* que suelen estar relacionados con nodos TOR y forman parte de la *Deep Web*.

Se accede a *hola.onion.city* en el navegador de la máquina *cliente*.

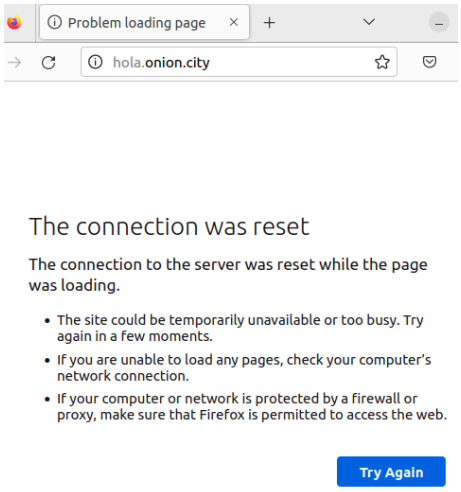


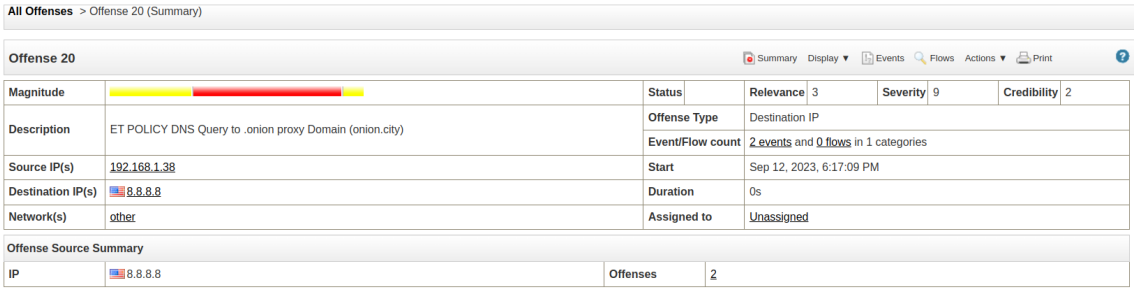
Figura 4.7: Acceso a *hola.onion.city*

Aparece el evento en el registro de actividades.

Event Name	Log Source	Event Count	Time ▲	Low Level Category	Source IP	Source Port	Destination IP	Destination Port
ET POLICY DNS Query to .onion proxy Domain (onion.city)	Suricata	1	Sep 12, 2023, 6:17:09 PM	Trojan Detected	192.168.1.38	33758	8.8.8.8	53
ET POLICY DNS Query to .onion proxy Domain (onion.city)	Suricata	1	Sep 12, 2023, 6:17:09 PM	Trojan Detected	192.168.1.38	52316	8.8.8.8	53

Figura 4.8: Evento *ET POLICY DNS Query to .onion proxy Domain (onion.city)*

La ofensa se genera.



All Offenses > Offense 20 (Summary)

Offense 20 — Summary | Display | Events | Flows | Actions | Print

Magnitude		Status		Relevance	3	Severity	9	Credibility	2
Description	ET POLICY DNS Query to .onion proxy Domain (onion.city)	Offense Type	Destination IP				
		Event/Flow count	2 events and 0 flows in 1 categories				
Source IP(s)	192.168.1.38	Start	Sep 12, 2023, 6:17:09 PM				
Destination IP(s)	8.8.8.8	Duration	0s				
Network(s)	other	Assigned to	Unassigned				

Offense Source Summary

IP	Offenses
8.8.8.8	2

Figura 4.9: Ofensa *ET POLICY DNS Query to .onion proxy Domain (onion.city)*

4.1.4. Resolución

Tras la comprobación de las reglas previas, y de otras que no se han documentado, se puede confirmar el correcto funcionamiento de Suricata así como su integración en QRadar.

4.2 OSSEC

4.2.1. Regla *Attempt to login using a non-existent user*

La regla genera ofensa cuando se intenta acceder al sistema con un usuario que no existe.

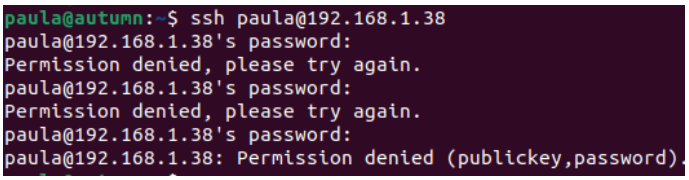
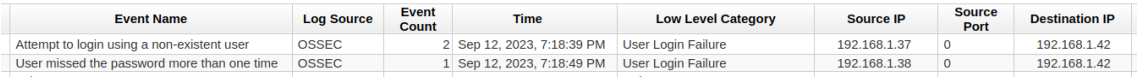


Figura 4.10: Intento de acceso con usuario inexistente

Se muestran los eventos asociados a este comportamiento.



Event Name	Log Source	Event Count	Time	Low Level Category	Source IP	Source Port	Destination IP
Attempt to login using a non-existent user	OSSEC	2	Sep 12, 2023, 7:18:39 PM	User Login Failure	192.168.1.37	0	192.168.1.42
User missed the password more than one time	OSSEC	1	Sep 12, 2023, 7:18:49 PM	User Login Failure	192.168.1.38	0	192.168.1.42

Figura 4.11: Regla *Attempt to login using a non-existent user*

Se genera la ofensa.

4.2.2. Regla *Root's crontab entry changed*

Esta regla detecta modificaciones en el *crontab*. Este fichero es delicado, ya que contiene todas las programaciones que son ejecutadas por el demonio *cron*.

Se añade como tarea en *crontab* la ejecución de un script que muestra por pantalla un listado de lo que hay en el escritorio.

All Offenses > Offense 23 (Summary)

Offense 23

Summary | Display | Events | Flows | Actions | Print

Magnitude		**Status**		**Relevance**	5	**Severity**	3	**Credibility**	2
Description	Attempt to login using a non-existent user	**Offense Type**	Source IP				
		Event/Flow count	3 events and 0 flows in 1 categories				
Source IP(s)	192.168.1.37	**Start**	Sep 12, 2023, 7:18:15 PM				
Destination IP(s)	192.168.1.42	**Duration**	23s				
Network(s)	Net-10-172-192.Net_192_168_0_0	**Assigned to**	Unassigned				

Offense Source Summary

IP	192.168.1.37	**Location**	Net-10-172-192.Net_192_168_0_0
Magnitude		**Vulnerabilities**	0
Username	Unknown	**MAC Address**	Unknown NIC
Host Name	Unknown		
Asset Name	Unknown	**Weight**	0
Offenses	5	**Events/Flows**	13

Figura 4.12: Ofensa *Attempt to login using a non-existent user*

```
root@cliente:/home/cliente# crontab -e
crontab: installing new crontab
root@cliente:/home/cliente#
```

Figura 4.13: Modificación crontab

```
 GNU nano 6.2                 /tmp/crontab.vSeHF9/crontab *
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
1 2 1 1 1 /home/cliente/Desktop/script.sh


^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

Figura 4.14: Fichero crontab

Se detecta la modificación.

Event Name	Log Source	Event Count	Time	Low Level Category	Source IP	Source Port	Destination IP	Destination Port
Root's crontab entry changed	OSSEC	1	Sep 12, 2023, 7:26:19 PM	Cron Status	192.168.1.38	0	192.168.1.42	0
Crontab opened for editing	OSSEC	1	Sep 12, 2023, 7:25:49 PM	Cron Status	192.168.1.38	0	192.168.1.42	0
Crontab opened for editing	OSSEC	1	Sep 12, 2023, 7:23:45 PM	Cron Status	192.168.1.38	0	192.168.1.42	0

Figura 4.15: Eventos asociados a la modificación del crontab

Se genera la ofensa con los eventos.

All Offenses > Offense 24 (Summary)

Offense 24 Summary Display ▾ Events Flows Actions ▾ Print

Magnitude		**Status**		**Relevance** 4	**Severity** 6	**Credibility** 4	
Description	Root's crontab entry changed	**Offense Type**	Source IP				
		Event/Flow count	3 events and 0 flows in 3 categories				
Source IP(s)	192.168.1.38	**Start**	Sep 12, 2023, 7:26:19 PM				
Destination IP(s)	192.168.1.42 Remote (2)	**Duration**	1s				
Network(s)	Multiple (2)	**Assigned to**	Unassigned				

Offense Source Summary

IP	192.168.1.38	**Location**	Net-10-172-192.Net_192_168_0_0
Magnitude		**Vulnerabilities**	0
Username	Unknown	**MAC Address**	Unknown NIC
Host Name	Unknown		
Asset Name	Unknown	**Weight**	0
Offenses	9	**Events/Flows**	246

Figura 4.16: Ofensa *Root's crontab entry changed*

4.2.3. Regla *New user added to the system*

Se crea un usuario por terminal desde la máquina del cliente.

```
cliente@cliente:~$ sudo useradd nuevoUsuario
[sudo] password for cliente:
cliente@cliente:~$
```

Figura 4.17: Creación de usuario *nuevoUsuario*

Se generan los eventos asociados a la actividad.

Event Name	Log Source	Event Count	Time	Low Level Category	Source IP	Source Port	Destination IP	Destination Port	Username
Login session opened	OSSEC	1	Sep 12, 2023, 7:35:10 PM	Session Opened	192.168.1.38	0	192.168.1.42	0	root
Successful sudo to ROOT executed	OSSEC	1	Sep 12, 2023, 7:35:10 PM	Privilege Escalation Succeeded	192.168.1.38	0	192.168.1.42	0	cliente
New group added to the system	OSSEC	1	Sep 12, 2023, 7:35:10 PM	Group Added	192.168.1.38	0	192.168.1.42	0	nuevoUsuario
Login session closed	OSSEC	1	Sep 12, 2023, 7:35:10 PM	Session Closed	192.168.1.38	0	192.168.1.42	0	root

Figura 4.18: Eventos de *New user added to the system*

Se genera la ofensa.

All Offenses > Offense 25 (Summary)

Offense 25 Summary Display ▾ Events Flows Actions ▾ Print

Magnitude		**Status**		**Relevance** 5	**Severity** 9	**Credibility** 3	
Description	New user added to the system	**Offense Type**	Username				
		Event/Flow count	1 events and 0 flows in 1 categories				
Source IP(s)	192.168.1.38	**Start**	Sep 12, 2023, 7:35:10 PM				
Destination IP(s)	192.168.1.42	**Duration**	0s				
Network(s)	Net-10-172-192.Net_192_168_0_0	**Assigned to**	Unassigned				

Offense Source Summary

Username	nuevoUsuario		
MAC Address	Unknown NIC	**Host Name**	Unknown
Last Known Host	Unknown	**Last Known Machine**	Unknown
Last Known MAC	Unknown	**Last Known IP**	Unknown
Last Observed	Unknown	**Last Known Group**	Unknown
Offenses	2	**Events/Flows**	2

Figura 4.19: Ofensa *New user added to the system*

4.2.4. Resolución

Tras la verificación de las reglas previas, y de otras que no se han documentado, se puede corrobora que OSSEC funciona correctamente y la integración en QRadar.

CAPÍTULO 5
Conclusiones y trabajos futuros

En este último capítulo, se realiza una reflexión sobre el proyecto expuesto con las conclusiones, se expresa la valoración a nivel personal y se muestra los trabajos futuros planteados para la continuación de este trabajo.

5.1 Conclusiones

Gracias al desarrollo de este proyecto, se ha podido desarrollar una plataforma SIEM para poder gestionar la seguridad en una empresa.

La integración de Suricata y OSSEC en QRadar facilita la detección de ataques en la máquina del cliente, agilizando el proceso de respuesta. Debido a las pruebas efectuadas, se ha podido comprobar el buen funcionamiento del sistema.

En cuanto a lo planificado, se han alcanzado todos los objetivos y se ha conseguido llevar a cabo todos los requisitos analizados y especificados para este proyecto.

5.2 Valoración personal

Como experiencia personal, el emprendimiento de este proyecto ha posibilitado:

- el aprendizaje de herramientas NIDS y HIDS, he comprendido cómo funcionan, cómo detectan las actividades sospechosas, cómo crear reglas personalizadas y soltarme con el uso de estas soluciones;
- el manejo de QRadar como admistradora, ha posibilitado cómo recoge los eventos, cómo reconoce de qué fuente procede la información, cómo se mapean los eventos, cómo se parsean las propiedades, cómo se crean y funcionan las reglas.

Este proyecto ha sido una experiencia enriquecedora a nivel personal, puesto que he podido enfrentarme a un proyecto desde cero, aplicando los conocimientos adquiridos durante el máster, y aprender nuevas herramientas con un resultado óptimo. Además, todo lo aprendido servirá en mi futuro como ingeniera en ciberseguridad.

5.3 Trabajos futuros

Una vez cumplimentados los objetivos de este proyecto, se presentan las siguiente propuestas para implementarlas en un futuro próximo:

- Añadir e integrar un firewall como fuente de datos.
- Instalar un antivirus e integrarlo en QRadar.
- Probar soluciones EDR.
- Investigar sobre alguna herramienta de ticketing para asociar con QRadar.

Bibliografía

[1] "Empresas españolas afectadas por ciberataques informáticos en 2023: ¿estás protegiendo tu negocio lo suficiente?" *Tekpyme*, 24-Jul-2023. [Online]. Disponible: https://es.linkedin.com/pulse/empresas-espa%C3%B1olas-afectadas-por-ciberataques-inform%C3%A1ticos-en. [Accedido: 12-Sep-2023].

[2] "La ciberseguridad desde los inicios: evolución de la seguridad" *INCIBE*, 29-Ago-2023. [Online]. Disponible: https://www.incibe.es/empresas/blog/la-ciberseguridad-desde-los-inicios-evolucion-de-la-seguridad. [Accedido: 12-Sep-2023].

[3] "¿Qué es un firewall?" *Cisco*. [Online]. Disponible: https://www.cisco.com/c/es_es/products/security/firewalls/what-is-a-firewall.html. [Accedido: 12-Sep-2023].

[4] "Network security" *Palo Alto Networks*. [Online]. Disponible: https://www.paloaltonetworks.com/network-security. [Accedido: 12-Sep-2023].

[5] "Next generation firewall (NGFW) - see top products" *Fortinet*. [Online]. Disponible: https://www.fortinet.com/products/next-generation-firewall. [Accedido: 12-Sep-2023].

[6] "Cisco Secure Firewall" *Cisco*. [Online]. Disponible: https://www.cisco.com/c/es_es/products/security/firewalls/index.html. [Accedido: 12-Sep-2023].

[7] "¿Qué es un sistema de detección de intrusos (IDS)?" *Check Point Software ES*, 15-Jul-2022. [Online]. Disponible: https://www.checkpoint.com/es/cyber-hub/what-is-an-intrusion-detection-system-ids/. [Accedido: 12-Sep-2023].

[8] "OSSEC" *OSSEC*, 23-Ene-2019. [Online]. Disponible: https://www.ossec.net/. [Accedido: 12-Sep-2023].

[9] R. Wichmann, "Samhain Labs" *La-samhna.de*. [Online]. Disponible: https://www.la-samhna.de/samhain/. [Accedido: 12-Sep-2023].

[10] "Security Onion Solutions" *Securityonionsolutions.com*. [Online]. Disponible: https://securityonionsolutions.com/. [Accedido: 12-Sep-2023].

[11] "Snort - network intrusion detection & prevention system" *Snort.org*. [Online]. Disponible: https://www.snort.org/. [Accedido: 12-Sep-2023].

[12] "Home," *Suricata*, 12-Ene-2021. [Online]. Disponible: https://suricata.io/. [Accedido: 12-Sep-2023].

[13] "The Zeek network security monitor" *Zeek*. [Online]. Disponible: https://zeek.org/. [Accedido: 12-Sep-2023].

[14] "¿Qué hace el antivirus para detectar el malware?" *INCIBE*, .30-May [Online]. Disponible: https://www.incibe.es/empresas/blog/hace-antivirus-detectar-el-malware. [Accedido: 12-Sep-2023].

[15] "Soluciones antivirus y seguridad en Internet" *Eset.com*. [Online]. Disponible: https://www.eset.com/es/. [Accedido: 12-Sep-2023].

[16] "Microsoft Defender" *Microsoft*. [Online]. Disponible: https://www.microsoft.com/es-es/security/business/microsoft-defender. [Accedido: 12-Sep-2023]

[17] "Antivirus, VPN, Protección de Identidad y Privacidad" *McAfee*. [Online]. Disponible: https://www.mcafee.com/es-es/index.html. [Accedido: 12-Sep-2023]

[18] "Detección y respuesta" *Trend Micro*. [Online]. Disponible: https://www.trendmicro.com/es_mx/business/products/user-protection/sps/endpoint/detection-response.html. [Accedido: 12-Sep-2023]

[19] "Soluciones de ciberseguridad de Kaspersky para hogares y empresas" *Kaspersky.es*. [Online]. Disponible: https://www.kaspersky.es/. [Accedido: 12-Sep-2023]

[20] "SentinelOne" *SentinelOne ES*, 22-Ene-2021. [Online]. Disponible: https://es.sentinelone.com/. [Accedido: 12-Sep-2023]

[21] "Web application firewall" *Barracuda Networks*. [Online]. Disponible: https://www.barracuda.com/products/application-protection/web-application-firewall. [Accedido: 12-Sep-2023]

[22] "ModSecurity" *GitHub*. [Online]. Disponible: https://github.com/SpiderLabs/ModSecurity. [Accedido: 12-Sep-2023]

[23] "All in one workspace solution for secure access to apps and data - citrix" *Citrix.com*. [Online] Disponible: https://www.citrix.com/. [Accedido: 12-Sep-2023]

[24] "Seguridad QRadar SIEM" *IBM*. [Online]. Disponible: https://www.ibm.com/es-es/products/qradar-siem. [Accedido: 12-Sep-2023]

[25] "AlienVault - open threat exchange" *AlienVault Open Threat Exchange*. [Online]. Disponible: https://otx.alienvault.com/. [Accedido: 12-Sep-2023]

[26] "Splunk" *Splunk*. [Online]. Disponible: https://www.splunk.com/. [Accedido: 12-Sep-2023]

[27] "How to install snort on Ubuntu" *UpCloud*, 20-Oct-2015. Disponible: https://upcloud.com/resources/tutorials/install-snort-ubuntu. [Accedido: 12-Sep-2023]

[28] "Snort Releases" *GitHub*. [Online]. Disponible: https://github.com/snort3/snort3/releases. [Accedido: 12-Sep-2023]

[29] "Fatal error: rpc/rpc.h: No such file or directory," *Ask Ubuntu*. [Online]. Disponible: https://askubuntu.com/questions/1360945/fatal-error-rpc-rpc-h-no-such-file-or-directory. [Accedido: 12-Sep-2023]

[30] "Rule Actions" *Snort*. [Online]. Disponible: https://docs.snort.org/rules/headers/actions. [Accedido: 12-Sep-2023]

[31] "Protocols" *Snort*. [Online]. Disponible: https://docs.snort.org/rules/headers/protocols. [Accedido: 12-Sep-2023]

[32] "Configuring Open Source SNORT" *IBM*, 08-May-2023. [Online]. Disponible: https://www.ibm.com/docs/en/dsm?topic=snort-configuring-open-source#t_dsm_guide_snort_cfg. [Accedido: 12-Sep-2023]

[33] "Suricata Releases" *GitHub*. [Online]. Disponible: https://github.com/OISF/suricata/releases. [Accedido: 12-Sep-2023]

[34] "3. Installation" *Suricata*. [Online]. Disponible: https://docs.suricata.io/en/latest/install.html. [Accedido: 12-Sep-2023]

[35] "Configuring Suricata to communicate with QRadar" *IBM*, 08-May-2023. [Online]. Disponible: https://www.ibm.com/docs/en/dsm?topic=configuration-suricata. [Accedido: 12-Sep-2023]

[36] "Zeek Releases" *GitHub*. [Online]. Disponible: https://github.com/zeek/zeek/releases. [Accedido: 12-Sep-2023]

[37] "Installing Zeek" *Zeek*. [Online]. Disponible: https://docs.zeek.org/en/v5.1.0/install.html#linux. [Accedido: 12-Sep-2023]

[38] "Zeek" *GitHub*. [Online]. Disponible: https://github.com/zeek/zeek. [Accedido: 12-Sep-2023]

[39] "Cluster Framework" *Zeek*. [Online]. Disponible: https://docs.zeek.org/en/v5.1.0/frameworks/cluster.html?highlight=cluster#zeek-cluster-setup. [Accedido: 12-Sep-2023]

[40] "Interactive Tutorial" *Zeek*. [Online]. Disponible: https://try.zeek.org/#/?example=hello. [Accedido: 12-Sep-2023]

[41] "OSSEC Releases" *GitHub*. [Online]. Disponible: https://github.com/ossec/ossec-hids/releases. [Accedido: 12-Sep-2023]

[42] "Installation requirements" *OSSEC*. [Online]. Disponible: https://www.ossec.net/docs/docs/manual/installation/installation-requirements.html#install-req. [Accedido: 12-Sep-2023]

[43] "Download OSSEC for Your Platform" *OSSEC*. [Online]. Disponible: https://www.ossec.net/download-ossec/. [Accedido: 12-Sep-2023]

[44] "Manager/Agent Installation" *OSSEC*. [Online]. Disponible: https://www.ossec.net/docs/docs/manual/installation/install-source.html. [Accedido: 12-Sep-2023]

[45] "While running ./install.sh getting this error usr/bin/ld: cannot find -lsystemd #1950" *GitHub*, 18-Feb-2021. [Online]. Disponible: https://github.com/ossec/ossec-hids/issues/1950. [Accedido: 12-Sep-2023]

[46] "Configuring OSSEC", *IBM*, 08-May-2023. [Online]. Disponible: https://www.ibm.com/docs/en/dsm?topic=ossec-configuring#t_dsm_guide_ossec_cfg. [Accedido: 12-Sep-2023]

[47] "Download Samhain", *Samhain Labs*. [Online]. Disponible: https://www.la-samhna.de/samhain/s_download.html. [Accedido: 12-Sep-2023]

[48] "Configuring syslog to collect Samhain events", *IBM*, 08-May-2023. Disponible: https://www.ibm.com/docs/en/dsm?topic=labs-configuring-syslog-collect-samhain-events#t_dsm_guide_samhain_syslog. [Accedido: 12-Sep-2023]

[49] "Try QRadar SIEM" *IBM*, 18-Jul-2016. [Online]. Disponible: https://www.ibm.com/community/qradar/ce/. [Accedido: 12-Sep-2023].

[50] "Download the IBM QRadar Community Edition" *IBM*. [Online]. Disponible: https://www.ibm.com/resources/mrs/assets/DirectDownload?source=swg-qradarcom&lang=es_ES. [Accedido: 12-Sep-2023].

[51] "IBM QRadar: QRadar Community Edition" *IBM*. [Online]. Disponible: https://www.ibm.com/community/qradar/wp-content/uploads/sites/5/2020/11/b_qradar_community_edition_7.3.3GA_v1.0.pdf. [Accedido: 12-Sep-2023].

[52] "Security Intelligence Tutorial, Demos & Uses Cases Version 329.pdf" *Jose Bravo*, 17-Ago-2023. [Online]. Disponible: https://ibm.ent.box.com/s/ich0yyiw54y0ek6s9a66xvtjku8e42rc/file/1283231547788. [Accedido: 12-Sep-2023].

[53] "UPDATED: A QRadar deploy changes on 31 December 2020 can impact product functionality," *IBM*, 09-Feb-2021. [Online]. Disponible: https://www.ibm.com/support/pages/node/6395080. [Accedido: 12-Sep-2023].

[54] R. Rojek, "Restart QRadar services" *Robert Rojek*, 10-Oct-2015. [Online][Online]. Disponible: https://www.robertrojek.pl/2015/10/10/restart-of-qradar-services/. [Accedido: 12-Sep-2023].

[55] "No logs/events seen" *IBM*. [Online]. Disponible: https://community.ibm.com/community/user/security/discussion/no-logsevents-seen. [Accedido: 12-Sep-2023].

[56] "Communication between agents and the OSSEC server" *OSSEC*. [Online]. Disponible: https://www.ossec.net/docs/docs/manual/agent/communication.html. [Accedido: 12-Sep-2023].

[57] "Managing Agents" *OSSEC*. [Online]. Disponible: https://www.ossec.net/docs/docs/manual/agent/agent-management.html. [Accedido: 12-Sep-2023].

[58] "Sending alerts via syslog" *OSSEC*. [Online]. Disponible: https://www.ossec.net/docs/docs/manual/output/syslog-output.html. [Accedido: 12-Sep-2023].

ANEXO

OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenible	**Alto**	**Medio**	**Bajo**	**No procede**
ODS 1. **Fin de la pobreza.**				X
ODS 2. **Hambre cero.**				X
ODS 3. **Salud y bienestar.**		X		
ODS 4. **Educación de calidad.**				X
ODS 5. **Igualdad de género.**				X
ODS 6. **Agua limpia y saneamiento.**				X
ODS 7. **Energía asequible y no contaminante.**				X
ODS 8. **Trabajo decente y crecimiento económico.**		X		
ODS 9. **Industria, innovación e infraestructuras.**		X		
ODS 10. **Reducción de las desigualdades.**				X
ODS 11. **Ciudades y comunidades sostenibles.**				X
ODS 12. **Producción y consumo responsables.**				X
ODS 13. **Acción por el clima.**				X
ODS 14. **Vida submarina.**				X
ODS 15. **Vida de ecosistemas terrestres.**				X
ODS 16. **Paz, justicia e instituciones sólidas.**				X
ODS 17. **Alianzas para lograr objetivos.**				X

Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.

Este TFM está enfocado a desarrollar una solución para proteger a las empresas de ciberataques. Estos ciberataques, en caso de ser efectivos y tener un impacto importante, pueden tener consecuencias desastrosas. Además del coste económico que supone un ataque efectivo, hay que tener en cuenta otros aspectos.
En el caso de ser algún centro sanitario, como un hospital, un ataque con *ransomware* en el que se dejan inoperativas la maquinaria y todos los dispositivos que se utilizan, puede tener implicaciones graves para los pacientes en estado crítico y con necesidades urgentes. Esto se enlazaría con el ODS de Salud y bienestar.
En el caso de ser alguna empresa de energía o de agua, un ciberataque puede afectar a la disponibilidad del servicio, haciendo que personas no tengan acceso a luz y agua, lo que se relaciona con el ODS de Industria, innovación e infraestructuras.
Para cualquier empresa, tanto los ataques como las consecuencias de estos, suponen unos grandes costes económicos. Esto se asocia al ODS de Trabajo decente y crecimiento económico.
Aunque sólo son 3 con los que se podría asociar, parte de estos objetivos como Hambre cero e Igualdad de género, entre otros, se tienen presentes en el día a día.