Implementación práctica de un Sistema de monitorización en tiempo real del tránsito en un centro de secundaria

Andrés Fraile Parra

Resumen– Debido a la rápida digitalización de todos los servicios, se nos presenta la necesidad de monitorizar para detectar trafico maliciosos en transito desde o hacia la organización. El objetivo de este trabajo es la instalación de un sistema de monitorización de red en un centro de secundaria, que nos permita observar dichos eventos. Para ello primero miraremos los componentes necesarios, siguiendo con la elección de la herramienta, en nuestro caso Security Onion, y finalizaremos con su instalación y configuración.

Palabras clave– Monitorización, Red, IDS, IPS, SIEM, SecurityOnion

Resum– A causa de la ràpida digitalització de tots els serveis, se'ns presenta la necessitat de monitoritzar per detectar trafic maliciosos en transit des de o cap a l'organització. L'objectiu d'aquest treball és la instal·lació d'un sistema de monitoratge de xarxa en un centre de secundària, que ens permeti observar aquests esdeveniments. Per a això primer comprovarem els components necessaris, seguint amb l'elecció de l'eina, en el nostre cas Security Onion, i finalitzarem amb la seva instal·lació i configuració.

Paraules clau– Monitorització, Xarxa, IDS, IPS, SIEM, SecurityOnion

Abstract– Due to the rapid digitization of all kinds of services, we are presented with the need to monitor to detect malicious traffic in transit from or to the organization. The objective of this work is the installation of a network monitoring system in a secondary school in order to observe these events. To do this we will first look at the necessary components, then we choose the tool, in our case Security Onion, and we will finish with its installation and configuration.

Keywords– Monitoring, Network, IDS, IPS, SIEM, SecurityOnion

1 Introducción - Contexto del trabajo

Cuando un dispositivo esta conectado a internet y hay uno o más servicios expuestos, siempre existen intentos de ataque, escaneos y búsqueda de vulnerabilidades. La función de un sistema de detección de intrusiones (IDS) sería detectar y registrar estos eventos, además de también observar las peticiones sospechosas que salen de nuestra organización.

- E-mail de contacto: andres.frailep@uab.cat
- Mención realizada: Tecnologías de la Información
- Trabajo tutorizado por: Lino de la Muñoza Muñoz (DEIC)
- Curso 2021/22

Este trabajo de final de grado (TFG) consiste en implementar un Sistema de Monitorización de tráfico, con un componente IDS/IPS.

Para ello hemos considerado que herramientas y que tipo de configuración usar, centrandonos principalmente en herramientas gratuitas y Open Source, decidiendo al final utilizar Security Onion.

2 Objetivos

El objetivo final del TFG es la implementación de un Sistema de Monitorización de tráfico. Para alcanzar dicho objetivo, dividiremos el trabajo en diferentes fases con objetivos propios para planificar el desarrollo.

Durante el curso del trabajo, se pretende seguir obteniendo información de múltiples fuentes, como documentación

y guías [1, 2], y de hacer pruebas en entornos virtuales. Para finalmente instalarlo en un entorno real donde comprobar su funcionamiento.

3 Estado del arte

Dentro del campo de la Monitorización de Redes nos encontramos que existen múltiples componentes que tener en cuenta, algunos incorporando funcionalidades de otros.

- Gestor de logs: Consisten en herramientas que reciben los datos que se van generando y los almacena y presenta de forma que se pueden consultar. Algunos ejemplos són Logstash y Graylog.
- Router: Se encarga de transmitir datos en diferentes redes. Un ejemplo es OpenWRT.
- Firewall: Se encarga de bloquear el trafico siguiendo unas reglas predefinidas o definidas por el usuario. Unos ejemplos serían pfSense y OPNsense.
- IDS/IPS (Intrusion detection/prevention system):

  Los sistema de detección/prevención de intrusos són un tipo de herramienta que se encargan de detectar eventos anormales y prevenir la intrusión de un agente externo utilizando reglas previamente definidas. Estas herramientas se pueden catalogar según la localización de la detección de eventos, como HIDS (Host IDS) o NIDS (Network IDS). Para más información ver [3, 4].

  Dentro de este grupo encontramos Snort, Suricata y Zeek, diferentes opciones con funcionalidad muy similar.
- SIEM (Security Information and Event Management):

  Los gestores de eventos e información de seguridad son herramientas más complejas que incluyen componentes IDS y de logs, entre otros, para poder visualizar y controlar los eventos que suceden. [5].

  Este es el grupo que nos hemos centrado más durante la implementación, porque cumplen con los objetivos. Un par de ejemplos son Security Onion y OSSIM.
- SOC (Security Operations Center):

  En el centro de operaciones de seguridad no solo se considera que incluye los componentes anteriores, sino que también se tiene en cuenta el equipo que se encargará de gestionar la red [6].

3.1. Posicionamiento del sistema

A continuación, una de las dudas que aparecen al querer implementar el sistema consiste en decidir donde y como colocarlo. Para ello, según el tipo de herramienta, existen dos opciones, colocarlo inline, con los puertos de red en modo puente (*bridge*), o configurar el en modo espejo (*mirror*) [7, 8].

- *Inline*

  El modo inline consiste en instalar el sistema en una máquina con dos tarjetas de red, configurándolo para monitorizar todo el tráfico que pasa a través de el. Este método tiene la ventaja de que no se necesita ningún sistema de conmutación especial, además de que te asegura la captura del 100 % de los paquetes y posible control sobre el qué trafico puede pasar. Con la desventaja de que necesita dos puertos de red, así como si está mal configurado y/o se cae, todos los dispositivos detrás del sistema se quedan sin acceso a internet.
- *Mirror*

  Por el otro lado, el modo espejo consiste en colocar el sistema, y configurar el switch, para transmitir todo el tráfico por el puerto espejo, donde el sistema recopila todos los paquetes. La ventaja de este método es que la máquina del sistema solo necesita un puerto red, y si deja de funcionar, los demás dispositivos siguen disponiendo de acceso a la red. La principal desventaja de este método es que no puede bloquear el trafico, solo monitorizarlo.

4 Herramientas Consideradas

En este apartado explicaremos un pequeño grupo de herramientas existentes relacionadas con la monitorización de red, y las pruebas que se hicieron con ellas. Finalmente, decidimos usar Security Onion

Estas herramientas serían OSSIM, pfSense/OPNsense y OpenWRT. Todas las pruebas han sido realizadas en un entorno virtual para comprobar su funcionamiento.

Como menciones honoríficas, comenzando con los SIEM, tenemos Apache Metron, debido a que el proyecto se ha retirado, Wazhu y MozDef.

4.1. OSSIM

Empezamos con OSSIM. Una potente herramienta de código abierto.

Tiene muchas funcionalidades muy bien integradas dento de una misma interfaz clara. En el panel principal tiene un dashboard editable, donde podemos observar gráficamente los diferentes aspectos de la herramienta, como por ejemplo el tipo de tráfico que fluye, los eventos generados y vulnerabilidades encontradas en la red.

Luego tiene apartados específicos para realizar análisis más detallados de los eventos generados, y programar búsquedas de vulnerabilidades.

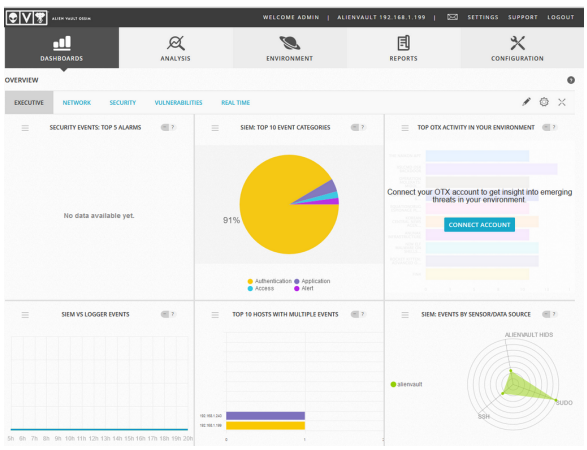

Fig. 1: OSSIM

4.1.1. Pruebas

En esta parte nos dedicamos a observar como se comporta OSSIM. Entre todas las funcionalidades que ofrece, comprobamos la vulnerabilidad de un dispositivo (en este caso una máquina virtual), usando nmap.

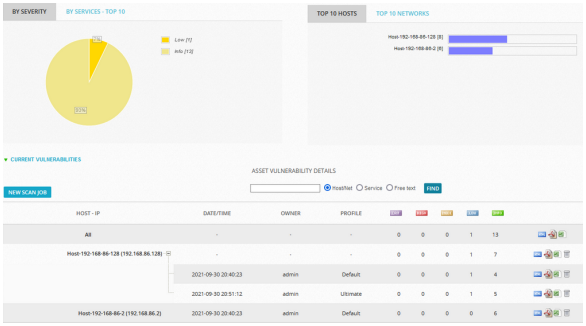

Fig. 2: Vulnerability Scan

4.1.2. Pros

- Cumple con todas las funciones que definimos como necesarias y más.
- Todas las funciones están bien integradas bajo una misma interfaz.
- Soporta plugins para añadir aún más funcionalidad.
- Fácil configuración de la pantalla principal para poder observar anomalías con mayor facilidad.
- Puedes instalar agentes/sensores en otros dispositivos para obtener más información.

4.1.3. Contras

- Algunas funciones están escondidas dentro de menús o solo aparecen con el botón derecho del ratón.

4.2. Maltrail

La siguiente herramienta considerada es Maltrail.

Maltrail es un sistema de detección de trafico malicioso que utiliza listas públicas de amenazas y eventos maliciosos. También es capaz (opcionalmente) de utilizar heurística para descubrir nuevas amenazas.

4.2.1. Pruebas

En este caso analizamos eventos generados obtenidos de una instancia de maltrail ya instalada en el centro.

Algunos ejemplos de resultados que obtenemos son:

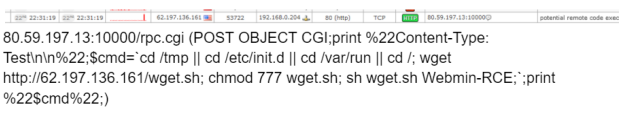

Fig. 3: Maltrail event 1

Donde podemos observar que un atacante ha intentado ejecutar comandos para descargar un script llamado wget.sh, y ha intentado darle permisos y ejecutarlo.

Tambien encontramos otro que periodicamente intenta obtener información de cuentas Microsoft:

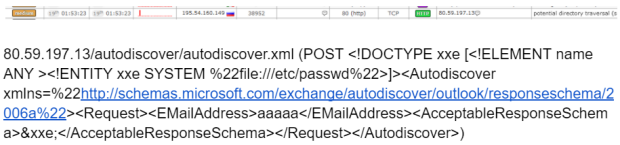

Fig. 4: Maltrail event 2

4.3. pfSense / OPNsense

Continuamos con pfSense y OPNsense.

Hemos agrupado estos dos Cortafuegos debido a que son prácticamente iguales con unas pequeñas diferencias. Ambos son Cortafuegos basados en FreeBSD, que cumplen la misma función con una interfaz ligeramente cambiada y soportan plugins para añadir funcionalidad.

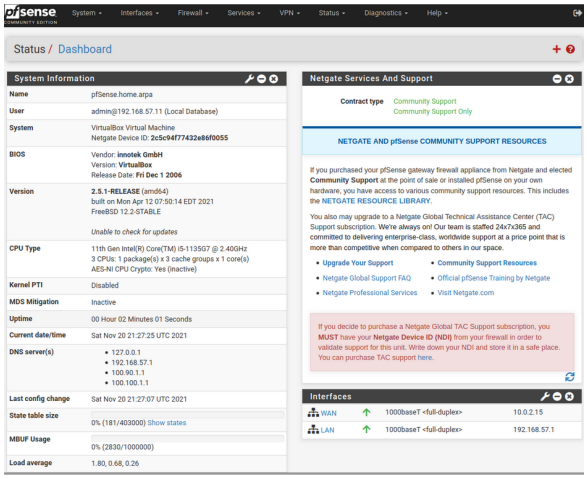

Fig. 5: pfSense

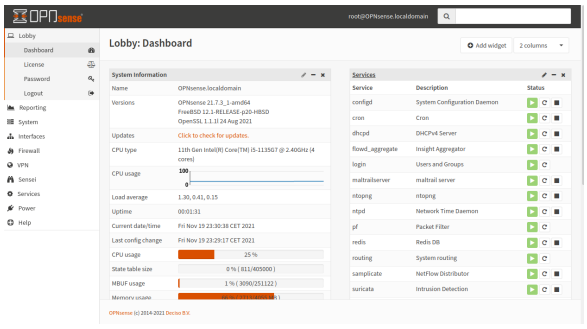

Fig. 6: OPNsense

4.3.1. Pruebas

En esta prueba empezamos con pfSense. pfSense no viene con ninguna utilidad IDS/IPS, pero con los plugins puede soportar todas la herramientas IDS consideradas (Suricata, Snort y Zeek). Para las pruebas decidimos usar Suricata, utilizando la siguiente guía [9].

Luego continuamos con OPNsense que viene integrado con Suricata por defecto. En este caso decidimos hacer las cosas diferentes y probar otros plugins. Instalamos maltrail,

explicado anteriormente, y ntopng que monitoriza el flujo de la red.

Cabe mencionar que después de la instalación, se notó la reducción del rendimiento de la máquina virtual.

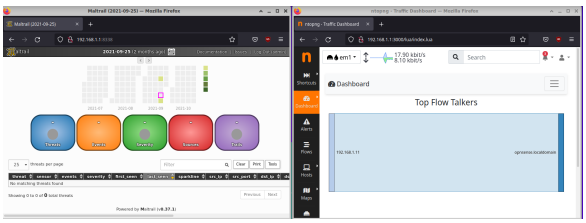

Fig. 7: Maltrail & Ntopng in OPNsense

En ambas opciones el sistema para ver los logs está limitado, por lo que, dependiendo de la cantidad de logs que se esperan, habría que aumentar la capacidad de disco, o bien se puede considerar instalar un gestor de logs externo.

4.3.2. Pros

- Los dos ofrecen un buen Cortafuegos sin mucha dificultad de configuración.
- Ambos ofrecen múltiples plugins que añaden funcionalidad interesante.

4.3.3. Contras

- Hay un número limitado de plugins, por lo que si no existe uno para una función concreta, lo más probable es que no puedas hacerla.
- Hay que tener en cuenta que y cuantos plugins se instalan en relación a los recursos que dispone la máquina.

4.4. OpenWRT

Terminamos con OpenWRT, un sistema operativo para routers basado en Linux.

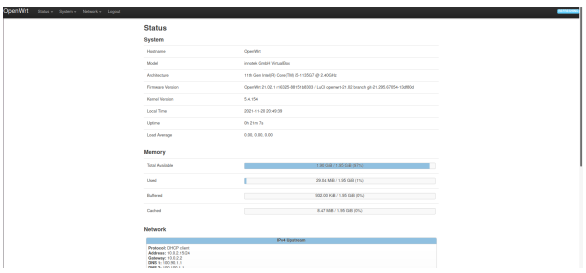

Fig. 8: OpenWRT

4.4.1. Pruebas

Al igual que otras opciones, OpenWRT acepta plugins que añade funcionalidad. Pero lo que lo hace brillar es que soporta Docker. Al soportar Docker se le puede añadir casi tanta funcionalidad como el dispositivo sea capaz de aguantar.

Para probarlo decidimos instalar maltrail desde Docker.

Respecto a los logs le sucede lo mismo que a la opción anterior, por lo que se pueden considerar las mismas soluciones.

4.4.2. Pros

- Es un buen SO para routers, basado en Linux.
- Tiene plugins, incluyendo Docker, por lo que puedes añadir toda la funcionalidad que se desea.

4.4.3. Contras

- Para añadir funcionalidad es obligatorio usar los plugins o contenedores Docker.
- Si se añade muchas funcionalidades, hay que tener cuidado con los recursos que consumen.
- Tiene la instalación más complicada [10].

5 Security Onion

Finalmente se ha decidido usar la herramienta Security Onion que nos ofrece todas las herramientas que necesitamos, además de la facilidad de encontrar información gracias a su documentación. Al igual que OSSIM, es un potente SIEM de código abierto.

5.1. Herramientas

Security Onion tiene incluida muchas herramientas integradas, como componentes IDS (Suricata y Zeek) y gestores de logs (Logstash), y extras que sirven para monitorizar el estado del equipo, para controlar múltiples dispositivos y agentes, y también para manejar tickets entre otras cosas.

Pero nosotros nos centraremos en las funciones principales que usaremos, con “Alerts” y “Hunt”, que son nativas de la interfaz web.

5.1.1. Alerts

Esta es la interfaz principal que usaremos. Desde esta interfaz se observan los eventos que han sucedido de manera superficial, para luego profundizar en la interfaz de caza.

Puedes hacer consultas (Query) predefinidas, como agrupar los eventos por tipo, gravedad y/o modulo que ha detectado el evento (Suricata, Zeek, ...), en un rango de tiempo determinado.

Aunque no te deja hacer Queries personalizadas directamente, es posible hacerlas a través de la URL, debido a como funciona (**https://(IP)/#/alerts?q=(Query)**, sin los paréntesis).

Para cada evento también es posible seleccionarlo para buscar solo dichos eventos, para excluirlos, o para investigarlos mas detalladamente con “Drilldown” o visualizarlo en la interfaz de caza.

5.1.2. Hunt

Continuamos con la interfaz de caza.

Esta interfaz es aparentemente similar a la anterior, en el sentido en el que se muestran los eventos. Sin embargo, está mas centrado en estudiar los eventos y visualizar gráficas que ayudan a visualizarlo.

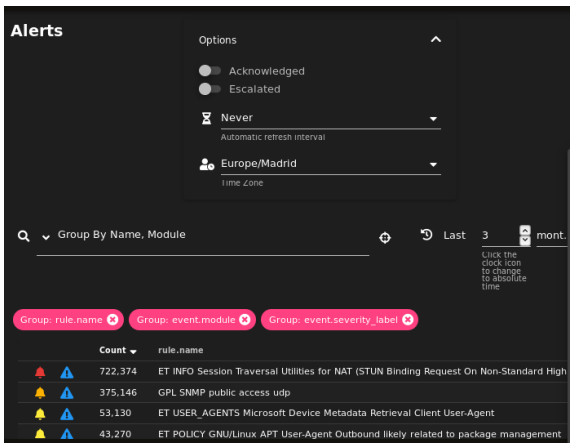

Fig. 9: Alerts Interface

Desde aquí se puede profundizar mucho más en cada evento para poder observar toda la información de un evento. También te permite importar un archivo Pcap, que contiene el tráfico realizado, para analizarlo y poder encontrar eventos.

Las opciones que ofrece al seleccionar los eventos son las mismas, excepto por el "Drilldown", que es exclusivo de la interfaz de alertas. También muestra más información por evento y te permite modificar directamente la Query desde la interfaz, y no es necesario modificar la URL.

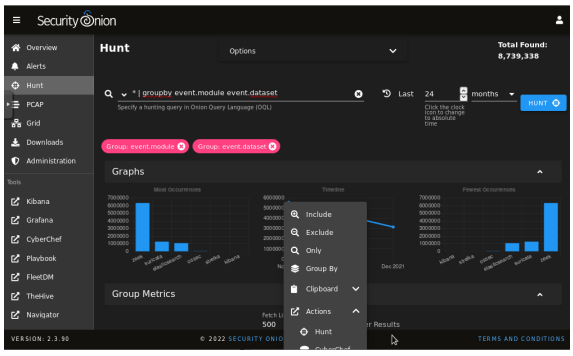

Fig. 10: Hunt interface

5.2. Instalación

La instalación de herramienta Security Onion se ha realizado en varios pasos.

El centro IES Sabadell proveyó de una máquina donde se instalaría Security Onion. Una vez instalado y confirmado su funcionamiento en mi hogar, se procedió a instalarlo en el Instituto.

Primero se comprobó usando la red de un aula que la herramienta captaba los datos que fluían por dicha red.

Posteriormente la máquina se instaló conectada a un switch que controla la salida del centro, para poder captar toda la información que entra y sale del mismo. Aunque tiene la limitación que el acceso a la red interna del instituto está detrás de una tabla NAT (Network Address Translation), por lo que no se puede saber el host implicado originario del centro.

También se modificó la interfaz de acceso del SecurityOnion para poder acceder a ella desde máquinas pertenecientes a esa red. Esto se intentó hacer mediante el archivo de

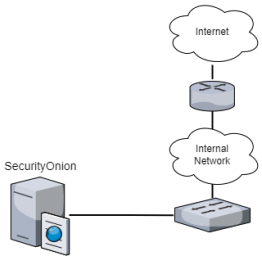

Fig. 11: SecurityOnion test location

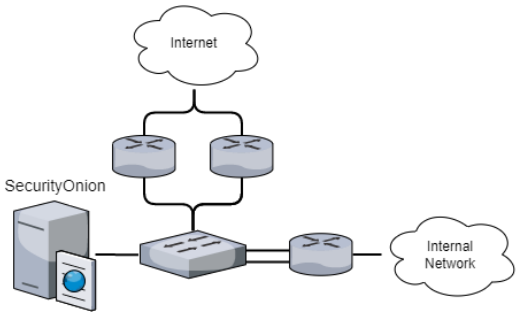

Fig. 12: SecurityOnion second location

configuración en */etc/sysconfig/network*, donde se modificó tanto la IP de la máquina como la puerta de enlace a la red. Además de ejecutar el comando *so-ip-update* para actualizar la IP de acceso a la interfaz web.

Finalmente se decidió instalar una segunda máquina con Security Onion en la red por donde accede todas las conexiones WiFi del centro, donde obtenemos más información

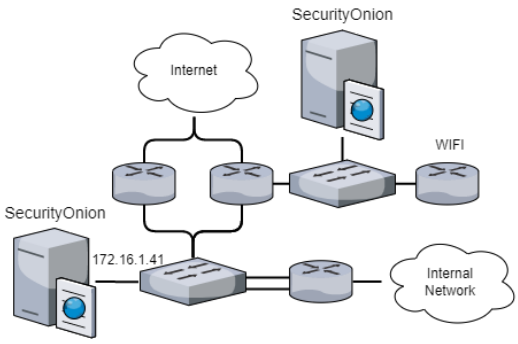

Fig. 13: SecurityOnion last location

5.3. Actividad

Con el sistema ya instalado hemos podido observar eventos que han sucedido en la red.

Primero hemos estudiado los eventos más comunes, que son los menos probables que sean reales, pero también son los más críticos para validar su veracidad. Principalmente observamos las direcciones IP de origen/destino, y buscaremos por Internet la localización de dicha IP para comprobarlos.

El evento más comuún, corresponde con la regla de Suricata. **SNMP public access udp**.

Para este evente se consultó con el tutor y se validó que es lícito, debido a que es el protocolo que utiliza el centro para monitorizar los switches y, aunque no fuera una falsa alarma, este evento tendría una fácil solución de bloquear

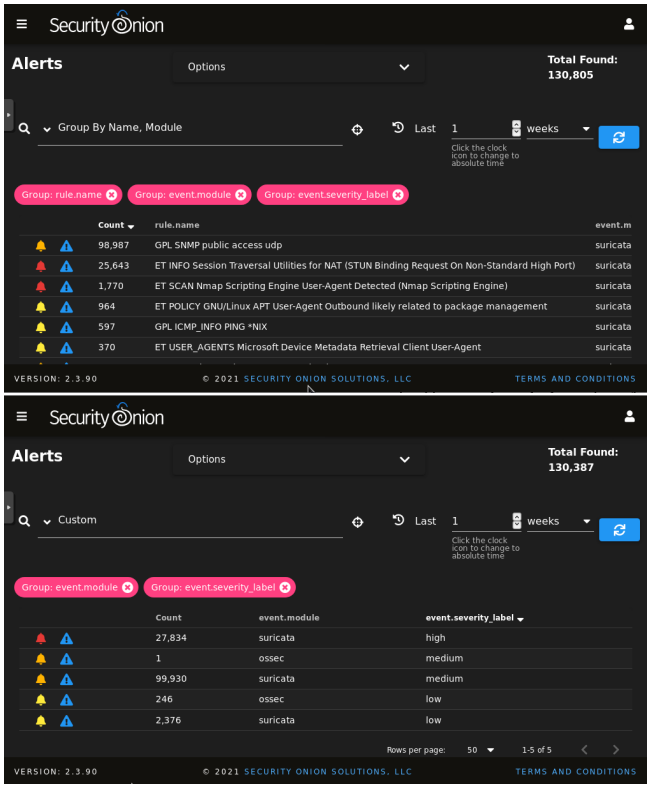

Fig. 14: SecurityOnion Alerts

el puerto 161 en el cortafuegos, que es el único puerto que usa.

Count	source.ip	destination.port
50,464	172.[redacted]	161
48,565	172.[redacted]	161
384	10.[redacted]	161
225	192.[redacted]	161
72	10.[redacted]	161
60	10.[redacted]	161
48	10.[redacted]	161
30	192.[redacted]	161
4	192.[redacted]	161

Fig. 15: SNMP by source.ip

El siguiente evento que encontramos es **ET INFO Session Traversal Utilities for NAT (STUN Binding Request on Non-Standard High Port)**.

En esta ocasión la IP de origen pertenece a dentro de la red interna, pero la de destino pertenece al exterior, aunque toda se encuentra en el mismo rango (109.200.198.0/24). Con esto descubrimos que estas direcciones pertenecen a la empresa i3D.net de los Países Bajos, que hostea varios servicios que utilizan WebRTC. Esto nos hace pensar que estas alarmas son falsos positivos, causados probablemente por alguien usando o dejándose abierto Discord o otra aplicación que use WebRTC. Más adelante se confirmó juntamente con el tutor que este uso es lícito debido a que los alumnos que se encuentran confinados hacen clases telemàticas a través de Discord.

Continuando con la búsqueda de eventos estudiamos ahora un escaneo Nmap, bajo el nombre **ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)**. Este ataque fue realizado desde el exterior para comprobar que esta herramienta realmente funcionaba.

Aparte de estos tres eventos que hemos estudiado, que eran los más comunes, existen muchos otros que pueden

Count	destination.ip
20,856	109.200.198.196
1,748	109.200.198.197
1,161	109.200.199.6
978	109.200.199.27
688	109.200.199.5
20	109.200.198.206
17	109.200.199.23
14	109.200.198.202
13	109.200.199.21
13	109.200.199.7
13	109.200.199.4
12	109.200.198.204
12	109.200.198.200

Fig. 16: STUN Binding Request

Count	source.ip
1,770	91.116[redacted]
59	10.1.5.100

Fig. 17: Nmap

ser peligrosos, incluyendo exploits, acceso a web de poca confianza y posibles filtraciones de información y archivos. Un ejemplo es el caso de **ET POLICY PE EXE or DLL Windows file download HTTP**, donde múltiples maquinas internas descargaron un .EXE o .DLL de una página que pertenece al dominio googleusercontent.com de Google Cloud, por lo que perfectamente podría ser malintencionado.

Otro ejemplo como los múltiples intentos originarios de Francia, China, Rusia y Romanía de aprovecharse, utilizando un exploit, de la recientemente descubierta vulnerabilidad de **log4j** de Java que ha afectado a un enorme número de empresas.

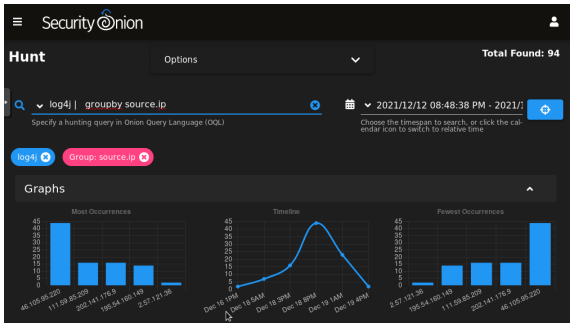

Fig. 18: Log4j

Aunque también existen muchos otros eventos que no tienen por que ser dañinos como la detección de Bing y Yandex Webcrawlers, inicios de sesión, existencia de trafico TOR, e información que el sistema operativo Windows envía a Microsoft.

5.4. Personalización

Una vez que hemos analizado cierta cantidad de tráfico durante un tiempo, nos damos cuenta de que se generan muchas alertas por los mismos eventos, independientemente de si son falsos positivos o no, y eventos que consideramos importantes pueden ser difíciles de encontrar entre todos los eventos. También consideraremos la opción de añadir alertas personalizadas.

5.4.1. Encontrar evento

Comenzamos con la dificultad de volver a encontrar un evento que hemos considerado importante.

Vamos a poner por ejemplo que nos interesa saber los usuarios que tienen una versión de java vulnerable.

Para ello se puede modificar la Query en la pestaña de alertas o caza, y buscar específicamente por java. Esta opción es muy sencilla de implementar y no se tarda mucho en ejecutar.

Aparte de esto, también es posible aprobar(símbolo de campana) y/o escalar(triangulo azul) dicho evento desde la interfaz de Security Onion. Esto puede tardar un tiempo en hacer efecto, pero una vez se aplica, podemos encontrar todos los eventos activando la opción de “Acknowledge” y/o “Escalated” en Opciones.

5.4.2. Ignorar eventos

Por otro lado, tenemos el problema contrario, de ignorar los eventos que hemos catalogado como falsos positivos. Si solo se quiere ignorar un número pequeño de eventos es posible hacerlo a través de Queries.

Sin embargo, si se quiere ignorar múltiples eventos, no tiene una solución tan sencilla.

En este caso es necesario modificar las reglas directamente desde el servidor, o bien con ssh (protocolo para acceder a una máquina remotamente). Existen múltiples modificaciones que podemos imponer a las reglas, como por ejemplo que ignoren ciertas IPs, pero para nuestro caso vamos a centrarnos en solo desactivarlas.

Para ello con **so-rule disabled add** se puede hacer de dos formas. Se puede ejecutar con ID de la regla, que es posible de obtener con la opción de “Drilldown”; y también es posible hacer con expresiones regulares, si queremos aplicarlo a múltiples reglas similares.

Como ejemplo vamos a deshabilitar la regla **ET INFO Session Traversal Utilities for NAT (STUN Binding Request on Non-Standard High Port)**, con id 2033078, usando **so-rule disabled add 2033078**, aunque también lo podríamos haber hecho como **so-rule disabled add ’re:STUN’**, aunque esto deshabilitaría todas las reglas que tuvieran “STUN” en su descripción.

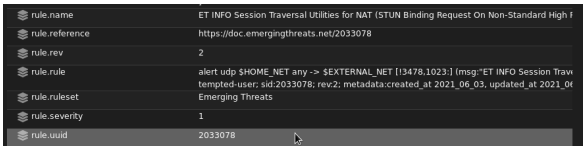

Fig. 19: STUN Suricata ID

5.4.3. Añadir regla

Finalmente, si queremos obtener alertas de algún evento en concreto, o queremos añadir algún exploit recientemente descubierto, consideramos la opción de añadir una regla propia.

Como tenemos que modificar reglas, al igual que en el caso anterior necesitamos tener acceso directo a la máquina.

Aquí se tiene que modificar el archivo de reglas locales, localizado en:

/opt/so/saltstack/local/salt/idstools/local.rules.

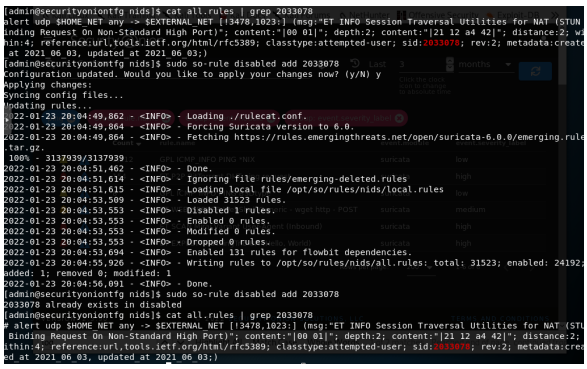

Fig. 20: Disable Rule

Donde el componente “Salt” lo usará para actualizar el archivo de donde lo lee Suricata cada 15 minutos. Esto es posible aplicarlo manualmente, siguiendo la guía en la documentación de Security Onion, en “Tuning”, en “Adding Local Rules”, pero no es necesario.

Como ejemplo vamos a añadir una regla que detecta cada vez que se genera una comunicación ssh en la red local definada para Suricata ($HOME_NET), que es definida durante la instalación y se puede modificar en el archivo de configuración de Suricata.

Una vez que la regla se añadido con éxito, podemos observar en la interfaz que realmente detecta la nueva regla.

Fig. 21: Custom Rule

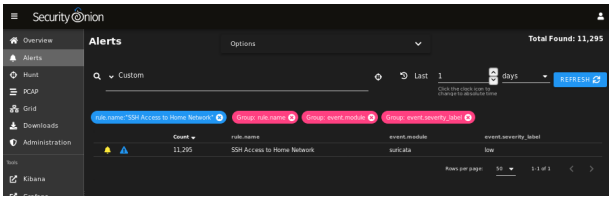

Fig. 22: Custom Rule in Interface

5.4.4. Instalación en modo inline

Finalmente se intentó configurar en modo inline desde un entorno virtual un Security Onion, con dos interfaces de red de monitorización.

Es necesario mencionar que Security Onion en su documentación indica que no esta pensado para funcionar en modo inline, no es recomendable y no ofrece ningún soporte.

En un principio se intentó sin éxito configurar el elemento IDS, en este caso Suricata, para que actuara como IPS, permitiéndonos aparte de observar el tráfico para generar alertas, bloquearlas.

Para realizarlo, se modificó el archivo de configuración de Surticata, especificando para las dos interfaces de red, las opciones *copy-mode:ips*, y *copy-iface:* especificando la otra interfaz.

Al no tener éxito se abandonó la idea de configurar el Suricata como IPS.

En este caso se decidió configurar las dos interfaces en modo puente, dejando el Suricata como IDS. Para ello, primero se creó la interfaz bridge en */etc/sysconfig/network-scripts/ifcfg-br0* siguiendo la guía [11] y añadiéndolo en las configuraciones de las redes implicadas.

Luego, para monitorizar la red creada, ejecutamos el comando *so-monitor-add*.

Con esto configurado, pudimos comprobar que los paquetes se transmitían por el puente, pero al ser probado en un entorno virtual no pudimos comprobar si funcionaba correctamente el componente IDS.

6 Conclusiones

Durante la realización de este proyecto se ha aprendido a instalar y a usar un sistema de monitorización de redes.

Ha sido un trabajo muy interesante, debido a que desconocía la gran cantidad de información que se puede obtener simplemente monitorizando la red, y todas las herramientas existentes para este motivo. He aprendido mucho instalando este sistema de monitorización.

Haciendo uso de la herramienta SecurityOnion hemos podido observar que en pocos días se generan cientos de miles eventos según las reglas por defecto. Ciertamente la gran mayoría de estos eventos resultan ser informativos, lícitos o falsos positivos; pero con la gran cantidad de eventos puede ser complicado realizar un buen análisis de la red sin personalizar las alertas.

Con todo lo observado podemos afirmar que este tipo de herramientas son actualmente imprescindibles por los peligros de los que nos puede llegar a proteger o, al menos alertar con suficiente tiempo para que no puedan causar daños muy graves, contolando el trafico en la red.

Esto lo podemos apoyar con el hecho de que los centros de cataluña están actualmente instalando Fortinet, una solución similar más potente pero pensada para empresas y no de codigo abierto.

Agradecimientos

Quisiera agradecer a mi tutor que ha hecho que sea posible la realización del trabajo. Así como también a mis padres por todo su apoyo recibido.

Referencias

[1] INCIBE. (2017, Nov.) Diseño y configuración de ips, ids y siem en sistemas de control industrial. [Online]. Available: https://www.incibe-cert.es/blog/diseno-y-configuracion-ips-ids-y-siem-sistemas-control-industrial

[2] A. Altvater. (2017, Jul.) Siem: A guide to successful implementation, strategy, and planning. [Online]. Available: https://stackify.com/siem-implementation-strategy-and-plan/

[3] Intrusion detection systems (ids) explained. [Online]. Available: https://cybersecurity.att.com/solutions/intrusion-detection-system/ids-explained

[4] N. Cavalancia. (2021, Feb.) Intrusion prevention systems explained: what is an ips? [Online]. Available: https://cybersecurity.att.com/blogs/security-essentials/intrusion-prevention-system-explained

[5] D. Kobialka. (2020, Jun.) What is a siem and what are the benefits for business? [Online]. Available: https://cybersecurity.att.com/blogs/security-essentials/siem-what-is-it-and-why-does-your-business-need-it

[6] M. Stone. (2021, Mar.) What is a security operations center (soc)? explaining the soc framework. [Online]. Available: https://cybersecurity.att.com/blogs/security-essentials/what-is-a-security-operations-center-soc

[7] M. Wilson. (2021, Jul.) Port mirroring – a definition & how it works (tutorial) & lab! [Online]. Available: https://www.pcwdld.com/port-mirroring-definition-and-tutorial

[8] S. Cooper. (2021, Apr.) What is port mirroring? the ultimate guide. [Online]. Available: https://www.comparitech.com/net-admin/ultimate-guide-to-port-mirroring/

[9] L. Systems. (2020, Aug.) Suricata network ids/ips installation, setup, and how to tune the rules & alerts on pfsense 2020. [Online]. Available: https://www.youtube.com/watch?v=S0-vsjhPDN0

[10] V. T. Corner. (2021, Jan.) Openwrt - vmware installation. [Online]. Available: https://www.youtube.com/watch?v=gJoTYxk-EYU

[11] D. Nanni. (2020, Nov.) How to configure a linux bridge interface. [Online]. Available: https://www.xmodulo.com/how-to-configure-linux-bridge-interface.html

[12] G. Blogger. (2020, May) Open source ids tools: Comparing suricata, snort, bro (zeek), linux. [Online]. Available: https://cybersecurity.att.com/blogs/security-essentials/open-source-intrusion-detection-tools-a-quick-overview

[13] K. Kent and M. Souppaya, *NIST Special Publication 800-92 : Guide to Computer Security Log Management*, May 2020. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-92.pdf

[14] Anton. (2019, Jun.) (very) basic elastic siem set up. [Online]. Available: https://haveyousecured.blogspot.com/2019/06/elastic-siem-set-up.html

[15] maysarax. (2019, Aug.) Security operations center -soc. [Online]. Available: https://github.com/maysarax/SOC

[16] S. Onion. (2021) Security onion documentation. [Online]. Available: https://docs.securityonion.net/en/2.3/index.html

[17] ——. (2021, Aug.) Quick malware analysis with security onion - malware-traffic-analysis.net pcap from 2021-08-05. [Online]. Available: https://www.youtube.com/watch?v=KBjr1fdb3jY