SIEM-SC: Análisis del coste de las políticas de seguridad en los eventos de un SIEM desde el punto de vista de la sostenibilidad

Juan Miguel López Velásquez
Facultad de Ingeniería
Istmo University (UNIS), Km 19.2
Fraijanes 01062, Guatemala
jmlopez@unis.edu.gt

Sergio Mauricio Martínez Monterrubio
dept. name of organization (of Aff.)
International de la Rioja University (UNIR)
La Rioja, Logroño, Spain
sergiomauricio.martinez@unir.net

Luis Enrique Sánchez Crespo
GSyA Research Group
University of Castilla-La Mancha (UCLM)
Ciudad Real, Spain
luisenrique.sanchez@unir.net

David Garcia Rosado
GSyA Research Group
University of Castilla-la Mancha (UCLM)
Ciudad Real, Spain
David.GRosado@uclm.es

***Abstract*—Actualmente la seguridad es cada vez más importante dentro de los sistemas de información empresariales. Además, toma cada vez más importancia aspectos como la sostenibilidad y el consumo energético asociados a los controles de seguridad. Por ello, es importante ser capaces de que los controles no sean solo seguros, sino también sostenibles. En el presente artículo se presenta una propuesta denominada SIEM-SC, para la construcción de un modelo que permite garantizar la privacidad en los registros obtenidos por un sistema SIEM, analizando dicho modelo no solo desde el punto de la preservación de la privacidad, sino también desde el punto de vista de la sostenibilidad. El desarrollo se ha realizado, analizando la información privada que contenían diferentes logs obtenidos mediante un Sistema de Gestión de Eventos e Información (SIEM por sus siglas en inglés), realizando previamente una formalización de los datasets utilizados, que ha permitido posteriormente un análisis sistematizado del consumo de recursos en diferentes dimensiones. Como conclusión, se ha demostrado la necesidad de contar con esa capa de seguridad adicional que permite garantizar la privacidad de los datos personales y que esa capa de seguridad tiene costes relevantes en el consumo de recursos según sea implementada de una forma u otra. En este documento también se concluyen propuestas futuras basadas en los hallazgos y errores en el proceso.**

***Index Terms*—Cumplimiento de la seguridad de la información, Eventos de seguridad, Información de seguridad, Leyes de privacidad, SIEM, GDPR**

I. Introducción

La seguridad en la tecnología de la información (TI) es un mundo en constante cambio que continúa evolucionando, transformándose y perfeccionándose a través del tiempo. El uso de diversas tecnologías para lograr un estado de seguridad general son amplios y se superan a sí mismos, de hecho, la tecnología en este campo nunca duerme. Una de las grandes áreas de la seguridad informática es el monitoreo: la idea de un ente de seguridad que pueda verlo todo, en todo momento y generar alertas cuando sea necesario es el objetivo final en esta área de seguridad. Dentro de este enorme paraguas de monitoreo continuo de segmentos de red, el tráfico de paquetes dentro de diferentes áreas de computadoras interconectadas, enfocándose solo en activos de alta prioridad y el estado real de segmentos de red es posible todo al mismo tiempo si un Sistema de Gestión de Eventos e información (SIEM por sus siglas en inglés) está configurado correctamente. El objetivo principal de un SIEM es centralizar todos los eventos de los dispositivos dentro de la red, los nuevos eventos entrantes mantienen actualizado el estado del dispositivo real y, en caso de una investigación post mortem, tenemos un gran repositorio para seguir las migas de pan para finalmente identificar una posible ruta de ataque/interrupción. Las principales características de la tecnología SIEM se pueden resumir de la siguiente manera: i) insustituible [1]; ii) proporciona una visión holística y reduce el riesgo a un nivel aceptable [2], [3]; iii) induce control crítico [4]; iv) proporciona un medio más fácil para ver el panorama general del ataque inicial y completa el ciclo de retroalimentación entre el monitoreo, el análisis y la defensa [5], [6]; v) fortalece la postura de seguridad, brinda la conciencia situacional requerida, lleva a cabo un análisis en tiempo real de las alertas de seguridad generadas por el hardware y las aplicaciones de la red [6]–[8], y vi) genera la solución más común y confiable, es crucial para la seguridad, también para los informes, auditorías de cumplimiento y la gestión de amenazas [9], [10].

La tecnología SIEM ha estado disponible en todos los sectores de la actividad humana que van desde la educación hasta la financiera, desde los gobiernos públicos hasta la medicina, es casi imposible nombrar una industria humana que no utilice este tipo de tecnología. Comenzó como un sistema de almacenamiento de registros centralizado que siguió evolucionando hacia la recopilación de eventos en tiempo

real junto con alertas generadas en tiempo real para notificar cualquier actividad anómala o sospechosa en cualquiera de los dispositivos bajo vigilancia. Esta es un área de tecnología de punta dado que se comunica con todas las tecnologías existentes, incluyendo diferentes versiones, marcas y protocolos. Este tipo de tecnologías son compatibles con más de 100 idiomas de diferentes tipos de dispositivos, incluidos: firewalls, proxies, servidores, cámaras, teléfonos, tabletas, enrutadores, conmutadores, etc.

La tecnología SIEM ha estado disponible durante más de 10 años, y en su vida útil se ha visto atrapada por diferentes cambios drásticos. Desde interfaces gráficas basadas en lenguajes nativos del sistema operativo hasta la explosión web, también poniéndose al día con los firewalls de próxima generación (NGFW por sus siglas en inglés) y finalmente con el mundo de los estándares formales de administración y gobernanza que intentan proteger la información capturada y utilizada por este tipo de tecnologías. Dentro de estos estándares podemos encontrar el Reglamento General de Protección de Datos (GDPR) que enumera pautas y procedimientos específicos para proteger los datos de Información Personal (PI por sus siglas en inglés).

Este tipo de estándares de seguridad se diseñaron y redactaron para proteger los datos de básicamente todos los sistemas informáticos que utilizarán PI. Como siempre, todas las tecnologías y sabores utilizados dentro de una organización tendrán que ponerse al día o sufrir las consecuencias generalmente relacionadas con costosas multas al fallar en auditorías. Una de estas tecnologías de la industria es el SIEM y este es el objetivo principal de este documento: diagnosticar una solución SIEM propuesta cuando aplica las pautas del GDPR para proteger la PI.

La solución SIEM propuesta se llama SIEM-SC (SIEM Secure Compliance por sus siglas en inglés); fue propuesta y aceptada por la comunidad científica en López et al [11]] y en resumidas cuentas el propósito total de esta arquitectura propuesta es abordar las necesidades de GDPR dentro de la tecnología SIEM.

El punto de referencia, los resultados y las comparaciones se llevarán a cabo teniendo en cuenta el uso del espacio (HDD), duración en segundos de las pruebas, los procesadores (CPU) y la memoria (RAM) utilizados por los contenedores en los diferentes escenarios a los que se exponen. Estos escenarios son propuestos principalmente por nuestros propios datasets incorporados construidos por el propio autor.

Estos datasets varían en tamaño, complejidad, contenido e información personal detallada en ellos. Mientras realizamos un seguimiento del rendimiento en las 4 categorías principales enumeradas en el párrafo anterior, analizaremos los resultados, propondremos mejoras y proveeremos el trabajo futuro sobre esta propuesta.

Para realizar estas pruebas utilizaremos distintas políticas para poder aplicar los aspectos exigidos por GDPR al encontrar eventos con PI. La habilitación y deshabilitación de estas políticas dará un abanico de resultados en cuanto a rendimiento y esto es lo que se busca presentar como hallazgo en esta investigación.

En el apartado II se encuentran los trabajos relacionados a éste, en el apartado III se encuentra un resumen sobre los datasets utilizados para realizar las pruebas sobre la propuesta. Más adelante en la sección IV los hallazgos y resultados serán detallados. La sección V contiene las conclusiones de esta investigación.

II. Trabajos relacionados

Al analizar muchas publicaciones y artículos diferentes que proponían una nueva ontología SIEM o marcos de trabajo novedosos relacionados a esta área de la tecnología encontramos varios. En este apartado analizaremos uno a uno para entender en que se centra su aporte y un símil con la propuesta detallada en esta investigación.

La publicación inicial analizada es González et al. [12], quien propuso una ontología unificada para asegurar la interoperabilidad entre múltiples sistemas y sus componentes: eventos, configuración, reacciones, detección y decisiones. El trabajo de Anumol et al. [13], quien diseñó un marco de trabajo que utiliza ML SVM (machine learning por sus siglas en inglés) para alimentar el proceso de toma de decisiones, centrándose únicamente en ataques de denegación de servicio distribuido (DDoS por sus siglas en inglés), Smurf y IP-sweep. La tecnología utilizada fue OSSIM AlienVault, y se establecieron las bases para este, aunque no se mencionaron resultados ni comparación alguna. Un trabajo inusual es el de Zhong et al. [14], quien sugirió una mejora desde la perspectiva humana con respecto a cómo un analista de seguridad (generalmente un ingeniero SOC) maneja la información proporcionada por las herramientas de monitoreo (incluyendo un SIEM) e intenta mejorarla con la máquina de estado de clasificación de datos sugerida bautizada DT-SM. En este experimento, 30 ingenieros de SOC fueron monitoreados mientras investigaban los registros de firewall (FW) y las alertas del sistema de detección de intrusos (IDS) mientras determinaban los patrones de ataque: mapeando sus acciones y construyendo una máquina de estado con la cual se volvería a analizar lo que había hecho el analista. Una vez esto se replicó se obtuvieron resultados extremadamente rápidos y con bajas tasas de falsos positivos. Según estos autores, ”La inteligencia humana, proporcionada por los analistas de ciberseguridad, está jugando un papel fundamental en los SOC para ’comprender’ las estrategias de ataque sofisticados a través del diagnóstico correlacionado avanzado”. Centrarse en la fase cognitiva humana de detección de problemas fue una verdadera innovación para este tema de investigación.

En su trabajo, Cinque et al. [15] propone un modelo con el que reconocer, comprender, analizar, agrupar y refinar la captura de datos en el entorno de Control de Tráfico Aéreo. Después de describir las múltiples áreas en las que se había implementado OSSIM AlienVault, se aplicó el escenario de modelado de datos. Después de comparar y analizar los resultados, los autores alientan a otros investigadores a refinar este nuevo tipo de métodos sintetizando los registros entrantes de áreas específicas de monitoreo del tráfico aéreo. Esta

propuesta es muy útil, aunque no todos los entornos cuentan con Control de Tránsito Aéreo que requiere monitoreo. Otro modelo es el de Bryant et al. [16], el cual se denomina Modelo Bryant Kill-Chain y consiste en un procedimiento de 7 pasos a seguir para detectarlos. Este modelo se probó en Logrhythm SIEM y se comparó con un entorno similar con el conjunto de reglas predeterminado (sin modificaciones). El modelo detectó el 96% de los 26 ataques, mientras que la configuración predeterminada (sin ediciones) detectó el 27%. Luego se propuso una ontología con la cual capturar delitos refinados: dentro de Logrythm y 120 clientes reales que le reportan, 22 ingenieros que trabajan las 24 horas del día, los 7 días de la semana en el SOC; sin modificación se detectaron más del 48% de las alarmas generadas ”estado crítico”. Junto con cuatro filtraciones de datos y 12 pruebas de penetración a ciegas: solo se identificaron el 50% de las pruebas de penetración. Sin embargo, la nueva ontología SIEM propuesta con la que agregar alertas y enviar correos electrónicos notificando acciones de respuesta (100 diarios a 1 diario) redujo el número de reglas de 128 a 39. Dado que solo se probó un tipo de SIEM, es necesario realizar una comparación más exhaustiva.

Estas propuestas enriquecen a la tecnología SIEM, cada paper contribuye a la expansión de esta tecnología. Las aportaciones científicas muestran y lideran las nuevas áreas de posible mejora para la tecnología SIEM, así como el ya mencionado SIEM-SC propuesto en López et al [11] .

III. Dataset

El dataset utilizado para realizar pruebas de estrés en SIEM-SC (detalles completos de esta propuesta pueden ser hallados en López et al [11]) se originó en varios entornos QRadar SIEM y la creación, edición, ajuste y modificación de este dataset CSV se encuentra a continuación.

El producto IBM QRadar pertenece a la tecnología SIEM, dentro del ámbito de la monitorización de la seguridad. Obtener acceso a múltiples entornos de QRadar con diferentes configuraciones, problemas y curiosidades a menudo es difícil y está restringido debido al acceso limitado a funciones que solo necesitan saber dado el objetivo principal de esta herramienta de seguridad: el monitoreo de una red.

El lapso de tiempo de los eventos analizados en este documento propuestos en los distintos datasets de QRadar es de 4 meses a 6 meses. Los registros de rastreo analizados son líneas de texto detallando acciones en varios módulos que difieren de las acciones simples del kernel (sesión iniciada, sesión finalizada) a tareas profundas de QRadar de backend. Dado que cada día se llena con miles de entradas, el dataset utilizado se enriquece con elementos que se pueden comparar como información personal: direcciones IP, nombres de host, FQDN y mucho más.

El archivo inicial analizado fue /var/log/messages, dado que estaba relacionado con el formato del kernel Red Hat Enterprise Linux (RHEL), se asumió que debería ser más fácil de digerir y por esto se inició con este archivo. El archivo analizado consistía en una salida de texto del sistema operativo (SO), cada entrada constaba de una línea dentro del archivo con varios formatos dentro. Debido a que el autor ya estaba acostumbrado a realizar scripts bash (.sh) como parte de una rutina diaria, decidimos usar este lenguaje de script para exportar todos los datos de los archivos originales a valores separados por comas (csv). El siguiente tipo de archivo analizado fue audit.log. El formato en este registro de auditoría /var/log/audit/audit.log cambia mucho, lo que complica la etapa de análisis general. Dentro de este archivo encontramos 11 formatos diferentes que no podían contenerse entre sí (únicos), por lo que los formatos para extraer la información fueron diferentes.

Finalmente, en las pruebas de estrés descritas y analizadas en las siguientes secciones de este documento, se utilizaron los siguientes datasets:

- audit.log del entorno virtual 15 con 885.574 eventos
- audit.log del entorno virtual 14 con 285.235
- messages de registro de seguimiento del entorno virtual A con 1.292.545

Dos de ellos forman parte del registro interno de QRadar y uno de ellos pertenece al entorno RHEL (el sistema operativo). En la siguiente sección detallaremos los resultados del SIEM-SC al capturar los ya detallados datasets. El dataset puede ser accesado y bajado desde este link: https://github.com/j-mlopez/QRADAR_dataset.git

A. Ambiente de pruebas

Los detalles sobre los contenedores se enumerarán en esta subsección. Se construyeron tres contenedores dentro de un CentOS, la máquina virtual tenía estas especificaciones: 4 núcleos, 32 GB de RAM y 16 GB de espacio distribuidos en varias particiones de núcleos. En este entorno se crearon 3 contenedores:

- El contenedor #1 es Análisis de datos. Esto se construyó sobre un contenedor acoplable que importa una versión especificada de Oracle Linux: 7 slim. Una vez que se creó esta instancia de Linux, se instaló httpd y httpd-tools. Otra herramienta importada fue la versión 8.0.19 de la suite mysql. Los puertos utilizados fueron 8090 externo, 80 interno para conectarse a Apache. La base de datos MySQL se configuró para escuchar en el puerto 3306. Todo esto desde un archivo de configuración Docker.
- El contenedor #2 es el controlador de eventos. Este contenedor se configuró desde la interfaz de línea de comandos (CLI por sus siglas en inglés) sin archivo Docker. Se utilizo Linux y Oracle incluido en la versión 7 slim. Luego se instaló httpd y httpd-tools suite para usar Apache. El puerto externo expuesto era el 8080 y el 80 interno para conectarse a Apache.
- El contenedor #3 es el controlador de claves. También desde la CLI, este contenedor se configuró con la imagen slim de Oracle 7. Este contenedor tenía una base de datos diferente para que residieran las claves de cifrado. El puerto interno y externo era el mismo: 3208.

Cada contenedor heredó las especificaciones del sistema operativo host. El uso de estos recursos se analizará en la

TABLE I
EXPRESIONES REGULARES USADAS COMO POLÍTICAS EN LAS PRUEBAS BETA DE SIEM-SC.

Nombre	Expresiones regulares
Email	^[\w\.\-=]+@[\w\.\-]+\.[\w]{2,3}$
Nº seguridad social	\b(?!000|666|9\d{2})([0-8]\d{2}|7([0-6]\d))([-]?|\s{1})(?!00)\d\d\2(?!0000)\d{4}\b
IPv4	^\d{1,3}[.]\d{1,3}[.]\d{1,3}[.]\d{1,3}$ \b
Nº tarjeta crédito	^(?:5[1-5][0-9]{2}|222[1-9]|22[3-9][0-9]|2[3-6][0-9]{2}|27[01][0-9]|2720)[0-9]{12}$
Nº tarjeta crédito 2	\b([4]\d{3}[\s]\d{4}[\s]\d{4}[\s]\d{4}|[4]\d{3}[-]\d{4}[-]\d{4}[-]\d{4}|[4]\d{3}[.]\d{4}[.]\d{4}[.]\d{4}|[4]\d{3}\d{4}\d{4}\d{4})\b
Nº tarjeta crédito 3	^3[47][0-9]{13}$
Código Zip	^((\d{5}-\d{4})|(\d{5})|([A-Z]\d[A-Z]\s\d[A-Z]\d))$
Dirección archivo	[^\\]+$
URL	(?i)\b((?:[a-z][\w-]+:(?:\/{1,3}|[a-z0-9%])|www\d{0,3}[.]|[a-z0-9.\-]+[.][a-z]{2,4}\/)(?:[^\s()]+|\(([^\s()]+|(\([^\s()]+\)))*\))+(?:\(([^\s()]+|(\([^\s()]+\)))*\)|[^\s`!()\[\]{};:\'\".,?]))
Todo encriptado	.*

TABLE II
PRUEBAS HECHAS A LOS DATASETS

Número de prueba	Escenarios	Dataset
24-27	A, B, C & D	audit.log recortado en el ambiente 15 con 400,000 líneas
29-32	A, B, C & D	audit.log recortado en el ambiente 15 with 600,000 líneas
33-36	A, B, C & D	audit.log en el ambiente 15
37-39	A, C & D	audit.log en el ambiente 14
40-42	A, C & D	messages en el ambiente 14
43-45	A, C & D	messages en el ambiente A

Sección IV mientras se analizan los eventos proporcionados a partir de los datasets en esta sección.

IV. RESULTADOS

Los resultados que se muestran en esta sección son parte de las pruebas de estrés de SIEM-SC. Las pruebas realizadas en el SIEM basado en contenedores tenían eventos leídos desde un dataset enviado directamente al contenedor dedicado del controlador de eventos a través del protocolo HTTP. Para obtener detalles sobre los datasets utilizados, consulte la sección III de este documento. Para obtener más detalles sobre el diseño y la configuración del SIEM-SC basado en contenedores, consulte la propuesta original en López et al [11].

Se propusieron varias expresiones regulares para proteger la información personal (PI) más común, todas ellas se describen en la Tabla I.

Con estas expresiones regulares variamos nuestros resultados habilitándolas todas, habilitando algunas y especialmente en la última descrita en la Tabla I: habilitada para que todos los eventos entrantes coincidan *ergo* sean encriptados. Con todas estas opciones de política disponibles, los diferentes tipos de pruebas que aplicamos se reunieron en 4 escenarios principales diferentes:

- Escenario A: todas las expresiones regulares que coinciden con la información personal están activas excepto la última (coincidir todas), comparadas y aplicadas en orden.

- Escenario B: todas las expresiones regulares que coincidan con cualquier evento en el escenario A deshabilitadas, el resto de ellas (sin coincidencia) habilitadas. En este escenario no se habilita la última política (cifrados todos).

- Escenario C: ninguna política de expresiones regulares activa. Lo que significa que recibiremos todos los eventos y los escribiremos directamente en nuestro contenedor sin ningún tipo de cifrado.

- Escenario D: última política especial que coincide con todos los eventos entrantes habilitada, el resto de políticas deshabilitadas.

Sin ningún tipo de evidencia se predijo que el escenario más complicado, más largo y más exigente de todos sería el D, ya que cifraba literalmente todos los eventos entrantes. En cuanto al más rápido, fácil y menos demandante sería el escenario C ya que ninguna política estaría habilitada.

Se ejecutaron 45 pruebas, las iniciales 17 fueron completamente irrelevantes y a manera de configuraciones iniciales para confirmar que las siguientes pruebas serían correctas y los datos monitoreados serían correctos. De la prueba #18 a la #45 obtuvimos hechos resaltables, los hallazgos y los puntos destacados representados en las siguientes figuras y párrafos pertenecen a esta fase de pruebas. De la prueba #18 hasta la #32 utilizamos datasets recortados, es decir las iniciales N cantidad de líneas de un dataset. Los escenarios A al D fueron probados en las pruebas #18 a la #32. De la #33 en adelante unicamente usamos los escenarios A, C y D ya que en base a lo observado no se encontró utilidad en monitoreo del escenario B en estas pruebas. En la Tabla II se enumeran los detalles de las pruebas, detallando a cual número de pruebas pertenece, a que escenario y a que dataset (o parte del dataset).

En general, obtuvimos 20 pruebas con datos exportados de carga promedio de CPU, uso de porcentaje de CPU, duración, espacio utilizado, porcentaje de memoria y utilización de memoria. Para cada escenario, 6 archivos CSV con datos que se pueden exportar como gráficos lineales o gráficos de barras; en total generamos 120 gráficos para todas las pruebas. En las siguientes imágenes seleccionadas y analizadas, intentamos comparar el rendimiento en varias áreas para que el lector pueda digerir y comprender el comportamiento general de SIEM-SC frente a diferentes datasets.

La Fig. 1 muestra la duración de las pruebas #24-#27 y #29-#32. La parte superior pertenece al dataset audit.log del entorno 15 recortado a 400,000 líneas con los 4 escenarios probados. La parte inferior pertenece a audit.log del entorno 15 recortado a 600,000 líneas con los mismos 4 escenarios.

La Fig. 2 muestra la duración de los datasets completos (sin recortar) audit.log de los entornos 14 y 15, así como messages del entorno 14 y A.

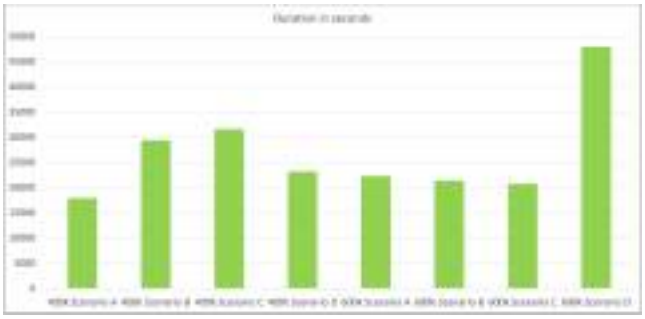

Fig. 1. Duración en segundos durante las pruebas #24-#27 y #29-#32 en dataset recortado de audit.log del ambiente 15.

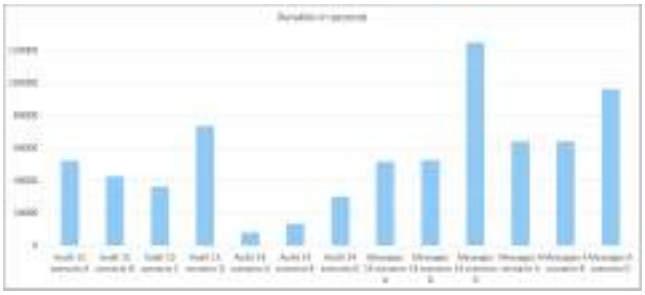

Fig. 2. Duración en segundos de las pruebas #33-#39 en los datasets audit.log del entorno 14 y 15. También pruebas #40-#45 en messages de los datasets del entorno 14 y A.

La Fig. 3 muestra el uso de espacio en megabytes al final de cada prueba, podemos ver que la utilización es mayor en el escenario D en el dataset recortado de 600,000, pero no en el dataset recortado de 400,000: en este, el mayor uso de espacio fue durante el escenario A.

Fig. 3. Espacio utilizado en megabytes (Mb) en las pruebas #24-#27 y #29-#32 en el dataset audit.log recortado del entorno 15.

La Fig. 4 compara la utilización del disco en las pruebas realizadas en datasets completos. La prueba de mayor recaudación en la categoría de espacio de estas pruebas fue el escenario D con el dataset audit.log del entorno 15 usando 194.12 megabytes. El más ligero en cuanto al espacio fue el escenario D con el dataset audit.log perteneciente al entorno 14: utilizando 115.77 megabytes.

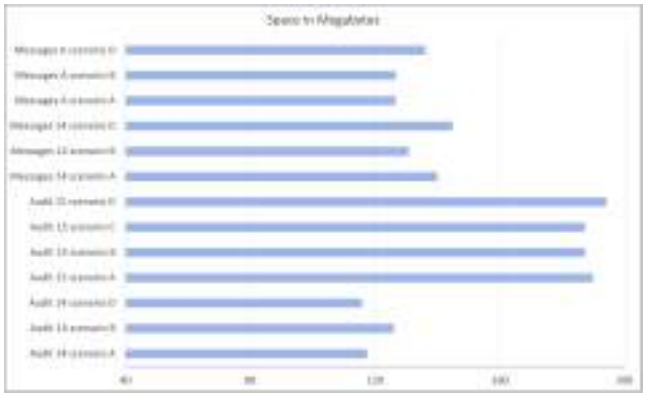

Fig. 4. Espacio utilizado en megabytes en pruebas #33-#45 en datasets completos: audit.log del entorno 14, audit.log del entorno 15, messages del entorno 14 y messages del entorno A.

La Fig. 5 compara el promedio de carga de CPU utilizado durante la ejecución de la prueba usando el escenario B y C con el dataset audit.log recortado de 400,000 líneas perteneciente al entorno 15. En la parte superior de la imagen, podemos ver la sobreutilización de CPU mientras que el escenario B esta siendo ejectuado comparado con un pico decente debajo del escenario C. El pico más alto del escenario B es 230.73 cuando en el escenario C el pico más alto es 48.46. Recordemos que el promedio de carga representa 3 gráficos, los cuales son el promedio de uso de CPU en: último minuto, últimos 5 minutos y últimos 15 minutos. La carga se recibe directamente de RHEL OS y, de acuerdo con su documentación, establece: ”El promedio de carga es un número que corresponde al número promedio de procesos ejecutables en el sistema” [17].

En la Fig. 6 comparamos el escenario C con el escenario D en cuanto a porcentaje de utilización de CPU, que presumiblemente sin ninguna evidencia asumimos que iban a ser los escenarios en los que se utilizarían menos y más recursos de CPU respectivamente. Estas comparaciones se realizaron durante las pruebas en las 600,000 líneas recortadas audit.log del dataset del entorno 15.

En la Fig. 7 , comparamos el dataset de registro messages del entorno 14 y A, ambos en el escenario D en cuanto a la utilización de CPU.

La última figura, Fig. 8, muestra el gráfico de resultados de monitorear el uso de memoria en megabytes para todos los datasets completos mientras se probaba el escenario D. Esta última prueba de esfuerzo nos mostró un pico de 401.3 Mb, cuando los otros 3 (en orden de izquierda a derecha, de arriba a abajo) son 643.6 Mb, 639.7 Mb y 651.3 Mb.

A. Análisis de hallazgos

Con un vistazo al monitoreo de los recursos, obtenemos una vista previa bastante amplia de cómo se realizaron, supervisaron las pruebas y el resultado para compararlas entre sí según los mismos principios: los mismos escenarios, los mismos datasets o la misma área: duración (segundos) o el uso del espacio (megabytes).

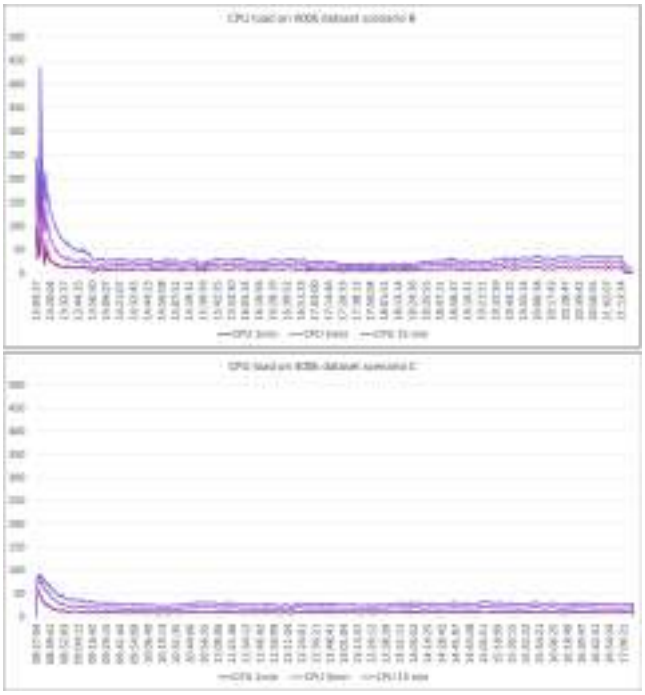

Fig. 5. Promedio de carga de CPU en comparación con los resultados de las pruebas #25 (escenario B) y #26 (escenario C) en el dataset audit.log con 400,000 líneas recortadas del entorno 15.

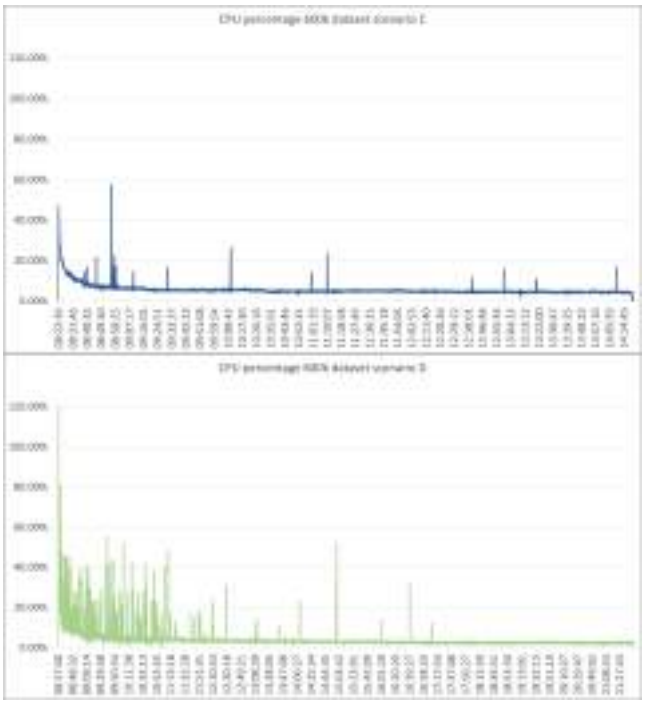

Fig. 6. Porcentaje de uso de CPU comparado como resultado de las pruebas #31 (escenario C) y #32 (escenario D) en el dataset recortado a 600,000 líneas de audit.log del entorno 15.

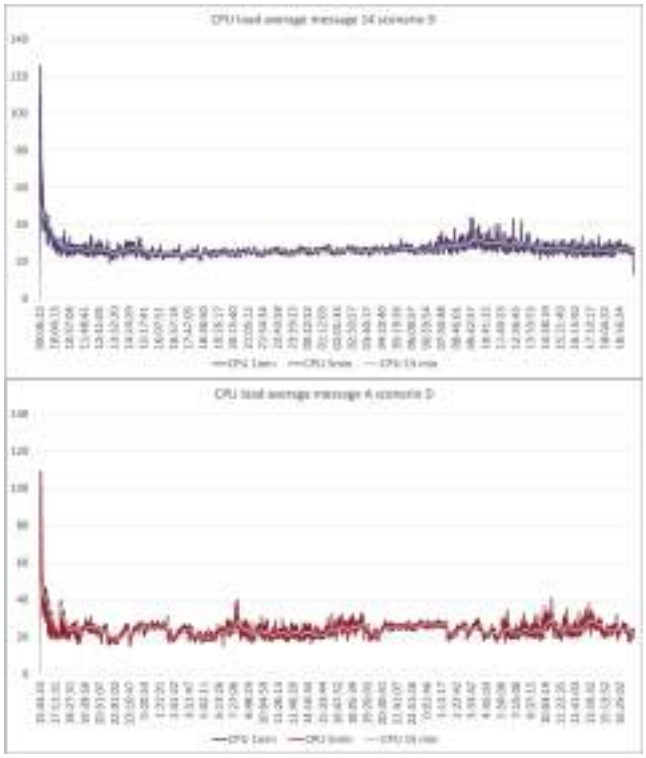

Fig. 7. Promedio de carga de CPU comparado con los resultados de las pruebas #42 y #45 del dataset messages completo del entorno 14 y A, ambos durante el escenario D.

Fig. 8. Utilización de memoria en megabytes de todos los datasets completos mientras se probaba el escenario D: audit.log del entorno 14, audit.log del entorno 15, messages del entorno 14 y messages del entorno A. Pruebas #36, #39, #42 y #45.

En Fig. 1 observe la cantidad de diferencia de tiempo en segundos del escenario D en ambos datasets: con 400,000 líneas terminadas en 23,221 segundos y con 600,000 líneas terminadas después de 47,961 segundos (más de 13 horas).

Así también en Fig. 2 en los 4 datasets completos, la mayor duración fue durante el escenario D. La prueba más rápida de todas fue el escenario A en audit.log del entorno 14 con un total de 8,030 segundos. La prueba más lenta tomó 124,734 segundos en el escenario D de messages pertenecientes al entorno 14 (más de 34 horas). Todo queda claro cuando comparamos la cantidad de líneas en cada dataset como se

indica en la Sección III. Nuestras predicciones apuntaban al escenario D como el más pesado y lento, las predicciones fueron correctas en todos los datasets completos, pero no en los datasets recortados, esto se muestra en la Fig. 3.

En Fig. 3 el dataset recortado de 400,000, la diferencia del escenario A usa 65.63 Mb como el más alto frente a los 34.48 Mb usados en el escenario D representado como más bajo. Con el dataset recortado de 600,000, obtuvimos el uso más alto en el escenario D con 72.91 Mb frente al más bajo en el escenario A con 65.47 Mb.

Antes de las pruebas, asumimos que el escenario D iba a desperdiciar más espacio, pero los resultados contradijeron nuestras predicciones sin fundamento, como podemos ver claramente en la Fig. 4.

Revisando la Fig. 5 curiosamente, los picos más pequeños van por encima de 30, pero el comportamiento ”normal” se mantiene por debajo de 30 en ambos escenarios.

En Fig. 6 podemos ver gráficamente más picos en el escenario D que en el C, lo que confirma nuestras suposiciones antes de las pruebas: el escenario D es más exigente con respecto a la CPU que el escenario C

Al revisar la Fig. 7 ambos gráficos nos muestran un comportamiento muy similar: un gran pico y luego un comportamiento ”normal” con picos más pequeños durante toda la prueba de esfuerzo.

La utilización de la memoria observada en Fig. 8 durante el escenario D fue mayoritariamente el mismo comportamiento: aumento inicial de la utilización de la memoria para normalizar y, finalmente, terminar la prueba. La única diferencia es que el pico inicial en 3 datasets que se muestran aquí son en su mayoría iguales, excepto audit.log del entorno 15.

V. Trabajo a futuro

Cada sistema informático construido hasta el día de hoy, partió de un diseño que era la base del mismo. Después de estos planes de diseño llegó a la parte de construcción, y después de construirlo se deben probar varios aspectos para declararlo utilizable. Una vez que se han identificado los problemas, la solución de estos problemas condujo a versiones más nuevas del sistema y, finalmente, se encuentran oportunidades para mejorar los objetivos generales del sistema y se aterrizan en una de estas nuevas versiones.

SIEM-SC se encuentra en sus primeras etapas de construcción y prueba, hasta ahora hemos centrado nuestro monitoreo durante las pruebas de estrés en la recepción de los eventos puntualmente en el contenedor del controlador de eventos, comparando cada evento con una política específica (expresión regular) y verificación de la aplicación el cifrado correspondiente. En las últimas etapas, centraremos nuestros esfuerzos en la fidelidad y confiabilidad de la herramienta, sabemos que se ignoraron algunos eventos en Apache, pero no pudimos explicar: cuántos, por qué y, sobre todo, cómo prevenir esta pérdida.

Por lo general, la tecnología SIEM depende del protocolo UDP para recibir una gran cantidad de información de diferentes fuentes a través del protocolo syslog (puerto 512), se sabe que UDP no confirma la entrega de ninguna información. Se tuvo en cuenta el mismo aspecto dentro del diseño de SIEM-SC, pero esto deberá cambiar y estaremos buscando esta funcionalidad en futuras versiones. Necesitamos un módulo de rendición de cuentas para proporcionar un resumen de cuántos eventos se enviaron, cuántos se recibieron y, si alguno falla, al menos una marca de tiempo de cuándo sucedió.

Una vez que el evento se captura y se escribe en SIEM-SC, ¿Cómo podemos usarlo? Una vez descifrado (si es necesario), ¿Dónde residirá? ¿Es SIEM-SC responsable de cualquier fuga de estos eventos descifrados una vez estén fuera de la plataforma y el rol responsable apruebe el acceso como se especifica en el GDPR (en tránsito)? ¿Se sobreexplotarán nuestros recursos al realizar esta etapa de descifrado? ¿Con qué frecuencia serán estos escenarios de acceso mediante descifrado en un lapso de 30 días? ¿Y si la ventana de monitoreo se reduce a 15 días?

Necesitamos un entorno de prueba en el que SIEM-SC se desenvuelva en eventos reales, escenarios reales para solicitar y aprobar el acceso a los datos de PI. Supervisar este entorno de prueba no sólo en el rendimiento, sino también en la usabilidad, la responsabilidad y la utilización de la solución propuesta. Todos los aspectos anteriores serán correctamente cubiertos en futuras versiones de SIEM-SC.

VI. Conclusiones

Este documento proporciona una amplia cantidad de aspectos relacionados con la seguridad en Tecnologías de la Información (TI), especialmente en el área de monitoreo. Dentro de esta investigación encontraremos una propuesta de solución que cubre especialmente el área de monitoreo enfocándonos en la tecnología SIEM y regulaciones internacionales como GDPR.

A lo largo de los años SIEM se ha ido adaptando para fusionar y superar otras tecnologías, ya que SIEM es un centralizador de eventos originados en diferentes sistemas tiene que poder entender lo que se le está enviando. La normalización de todos los lenguajes de seguridad es un desafío, especialmente debido al hecho de que la tecnología monitoreada sigue cambiando y transformándose en nuevos productos, nuevas versiones y nuevos desafíos. Dentro del SIEM, las diferentes demandas de mejoras residen en las mejoras tecnológicas reales del producto, como nuevas funcionalidades, presentación de los datos al usuario y, por supuesto, la compatibilidad de la herramienta como tal. Dentro de las normativas internacionales podemos encontrar una gran variedad de estas que afectan, regulan y exigen medidas extra sobre el funcionamiento de un producto incluyendo la tecnología SIEM.

En la solución exhaustivamente probada en esta investigación, bautizada SIEM-SC (Cumplimiento Seguro por sus siglas en inglés), tomamos una de estas regulaciones internacionales y aterrizamos los requisitos en aspectos funcionales de la solución cumpliendo con la protección requerida por GDPR. Desarrollamos una solución basada en contenedores en la que podemos ajustar y modificar las políticas aplicadas

al tráfico entrante para detectar información personal. Cifrar la información personal una vez hallada, al recibir los eventos, dentro de SIEM-SC al aplicarse las políticas utilizando: expresiones regulares, contenedores docker, datasets y Apache.

Durante las pruebas de estrés, monitoreamos los recursos asignados a los diferentes contenedores pertenecientes a SIEM-SC mientras se capturaban, analizaban y almacenaban los eventos entrantes reales dentro de la solución. Centramos nuestros 4 datasets en una batería de 45 pruebas en 4 escenarios diferentes. Estos escenarios consistieron en habilitar y deshabilitar nuestras 10 políticas propuestas que consisten en expresiones regulares para detectar una de estas categorías de información personal (PI) dentro de los eventos entrantes: correo electrónico, número de seguro social, direcciones IPv4, tarjetas de crédito, códigos postales, rutas de archivos y URLs.

Estos 4 escenarios se tipificaron como: escenario A con todas las políticas habilitadas, escenario B deshabilitando las políticas coincidentes mientras se usaba el escenario A, escenario C deshabilitando todas las políticas (todo el texto entrante se guarda como texto plano sin protección alguna) y escenario D habilitando una política especial que encripta todos los eventos recibidos. El resultado de monitorear estas 45 pruebas fue una cantidad inmensa de datos ya que mientras se ejecutaban las pruebas monitoreamos 4 áreas base diferentes: CPU, RAM, espacio y duración. De los 4 escenarios, llegamos a la conclusión de que el uso excesivo de recursos más pesado, más prolongado y más abundante se produjo en el escenario D, que aplicó la política "todos los eventos entrantes se cifran" mediante esta expresión regular: .*

La tecnología SIEM necesita este tipo de propuestas, como SIEM-SC, para impulsar herramientas que se podrían utilizar en los productos SIEM. Esto para materializar lo que se puede hacer y lo que se necesita solicitado por el mercado, los usuarios o, en este caso, una regulación internacional (GDPR). Esto tiene como objetivo forzar buenas prácticas y ofuscación dentro de un producto de seguridad solo para ocasiones especiales, como en SIEM-SC: información personal almacenada dentro de la solución.

ACKNOWLEDGMENT

Este trabajo es parte de los proyectos ALBA-UCLM (TED2021-130355B-C31) financiado por MCIN/AEI/10.13039/501100011033/Unión Europea NextGenerationEU/PRTR, AETHER-UCLM (PID2020-112540RB-C42/ AEI/10.13039/501100011033), and MESIAS (2022-GRIN-34202) financed by FEDER.

REFERENCES

[1] O. Podzins and A. Romanovs, "Why SIEM is Irreplaceable in a Secure IT Environment?" in *2019 Open Conference of Electrical, Electronic and Information Sciences (eStream)*. IEEE, apr 2019, pp. 1–5.

[2] D. A. Fernandes, L. F. Soares, J. V. Gomes, M. M. Freire, and P. R. Inácio, "A Quick Perspective on the Current State in Cybersecurity," *Emerging Trends in ICT Security*, pp. 423–442, 2014.

[3] S. Snedaker and C. Rima, "Chapter 7 - business continuity/disaster recovery plan development," in *Business Continuity and Disaster Recovery Planning for IT Professionals*, second edition ed. Boston: Syngress, 2014, pp. 369–411.

[4] H. Dalziel, "Chapter 15 - information security incident management," in *Infosec Management Fundamentals*, H. Dalziel, Ed. Boston: Syngress, 2015, pp. 45–46.

[5] A. Liska and G. Stowe, *DNS network security*. Elsevier, 2016, pp. 93–119.

[6] E. Knapp, "Chapter 11 common pitfalls and mistakes," in *Industrial Network Security*, E. Knapp, Ed. Boston: Syngress, 2011, pp. 303–312.

[7] A. K. Sood and R. Enbody, "Chapter 8 - challenges and countermeasures," in *Targeted Cyber Attacks*, A. K. Sood and R. Enbody, Eds. Boston: Syngress, 2014, pp. 123–134.

[8] V. Casola, A. De Benedictis, M. Rak, and U. Villano, "Chapter 11 - monitoring data security in the cloud: A security sla-based approach," in *Security and Resilience in Intelligent Data-Centric Systems and Communication Networks*, ser. Intelligent Data-Centric Systems, M. Ficco and F. Palmieri, Eds. Academic Press, 2018, pp. 235–259.

[9] J. Lee, J. Kim, I. Kim, and K. Han, "Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles," *IEEE Access*, vol. 7, pp. 165 607–165 626, 2019.

[10] E. Bertino, "Data Protection from Insider Threats," *Synthesis Lectures on Data Management*, vol. 4, no. 4, pp. 1–91, jun 2012.

[11] J. M. Lopez, "Systematic review of SIEM technology: SIEMSC birth," *International Journal of Information Security*, no. 22, p. 691–711, jan 2023.

[12] G. Gonzalez Granadillo, H. Débar, G. Jacob, C. Gaber, and M. Achemlal, "Individual countermeasure selection based on the return on response investment index," in *Computer Network Security*, I. Kotenko and V. Skormin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 156–170.

[13] E. T. Anumol, "Use of Machine Learning Algorithms with SIEM for Attack Prediction," in *Intelligent Computing, Communication and Devices*, L. C. Jain, S. Patnaik, and N. Ichalkaranje, Eds. New Delhi: Springer India, 2015, pp. 231–235.

[14] C. Zhong, J. Yen, P. Liu, and R. F. Erbacher, "Automate Cybersecurity Data Triage by Leveraging Human Analysts' Cognitive Process," in *2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS)*. IEEE, apr 2016, pp. 357–363.

[15] M. Cinque, R. Della Corte, and A. Pecchia, "Contextual filtering and prioritization of computer application logs for security situational awareness," *Future Generation Computer Systems*, vol. 111, pp. 668–680, oct 2020.

[16] B. D. Bryant and H. Saiedian, "Improving SIEM alert metadata aggregation with a novel kill-chain based classification model," *Computers & Security*, vol. 94, p. 101817, jul 2020.

[17] R. Customer portal, "Monitoring CPU Utilization on Red Hat Enterprise Linux," 2023, accessed: 2023-07-10. [Online]. Available: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/4/html/introduction_to_system_administration/s2