UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

VICERRECTORÍA DE TECNOLOGÍA

DIRECCIÓN DE SERVICIOS TELEMÁTICOS Y REDES

MARCO DE TRABAJO PARA LA GESTIÓN CENTRALIZADA DE TRAZAS DE SEGURIDAD UTILIZANDO HERRAMIENTAS DE CÓDIGO ABIERTO

Tesis presentada en opción al título de Máster en Informática Aplicada

Autor: Ing. Joelsy Porven Rubier

Tutor: Dr C. Raydel Montesino Perurena

La Habana

Julio de 2015

A mis padres Pucha y Miguel Angel.

A mi hermano Jose Miguel y a mi sobrino miguelito.

Agradecimientos

En especial deseo agradecer Dr C Raydel Montesinos, tutor, compañero de trabajo y amigo, quien de forma muy acertada me orientó en el desarrollo de este trabajo. Además quiero agradecer a Jissie Vaquero, quien es una persona muy especial para mí. A David Silva, amigo de mucho tiempo y de varias empresas. A Yanicet, Pablo y a todo el grupo de servicios telemáticos y al grupo de redes de la UCI.

A todos los que me apoyaron y contribuyeron de una forma u otra a que este trabajo fuera posible, mi más sincero agradecimiento.

Declaración jurada de autoría

Declaro por este medio que yo, Joelsy Porven Rubier con carné de identidad 79082801366, soy el autor principal del trabajo final de maestría Marco de trabajo para la gestión centralizada de trazas de seguridad utilizando herramientas de código abierto, desarrollado como parte de la Maestría en Informática Aplicada.

El presente trabajo fue desarrollado individualmente en el transcurso de los años 2013-2015

Finalmente declaro que todo lo anteriormente expuesto se ajusta a la verdad, y asumo la responsabilidad moral y jurídica que se derive de este juramento profesional.

Y para que así conste, firmo la presente declaración jurada de autoría en La Habana a los ____ días del mes de _____ del año 2015.

_____

Firma del maestrante

Resumen

Los sistemas de gestión de trazas son comúnmente subutilizados, generalmente se instalan para garantizar el cumplimiento de políticas y no se aprovechan los datos generados en el proceso de mejoras de los controles de seguridad. La mayoría de las instituciones recolectan y almacenan las trazas, presentando su mayor dificultad en la extracción de la información que pueda ser procesable.

El proceso de gestión de trazas de seguridad requiere de variantes que integren la variedad de formatos existentes, garantizar la normalización de la información generada, su almacenamiento, además de contar con herramientas para la búsqueda y visualización de los datos.

En la presente investigación se define un marco de trabajo para la gestión centralizada de trazas de seguridad que facilite la búsqueda, procesamiento y visualización de la información generada y contribuya a disminuir el tiempo de búsqueda sobre los registros almacenados. Como parte del desarrollo del marco de trabajo se hace un análisis de las referencias existentes a la gestión de trazas en las normas, guías y documentos de buenas prácticas más importantes, seleccionando los principales componentes y eventos que deben ser registrados. Para su implantación, se propone una arquitectura de despliegue donde se seleccionan las herramientas de código abierto necesarias.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

Introducción

El aumento acelerado de la informatización de la sociedad trae consigo la dependencia cada vez mayor de los sistemas informáticos. En la actualidad una gran parte de la información se encuentra en soporte digital. En correspondencia con este desarrollo también se puede apreciar un aumento acelerado de los incidentes de seguridad.

La información personal de los usuarios se almacena casi completamente en formato digital. Día a día crece aceleradamente el número de empresas donde su activo fundamental es la información disponible en bases de datos y soporte digital. Actualmente puede incluso perderse la noción de la localización física de los datos si se utilizan servicios de computación en la nube, lo que incluye nuevos retos en cuanto a la seguridad (Rittinghouse y Ransome, 2009)

Junto con el incremento de la información en línea también se incrementan los ataques y la ocurrencia de brechas de seguridad. Con el objetivo de monitorizar el funcionamiento de los sistemas de comunicaciones y servicios, detectar y prevenir incidentes de seguridad, cumplir con políticas establecidas y elaborar reportes, se han desarrollado un gran número de herramientas destinadas a procesar las trazas generadas por aplicaciones y dispositivos. Estas herramientas pueden catalogarse de sistemas para la gestión de trazas (LM por sus siglas en inglés) y sistemas de gestión de eventos de Información de seguridad (SIEM por sus siglas en inglés).

Los sistemas SIEM ofrecen una solución completa para la gestión de eventos de seguridad, pero no son factibles para muchas empresas debido a los elevados costos de licencias para adquirir el producto. Por otra parte, un sistema SIEM en sí mismo no garantiza ningún tipo de seguridad y requiere de personal especializado que lo ajuste al entorno donde se va a desplegar. Actualmente muchos de los productos SIEM oscilan en precios que equivalen a contratar ingenieros dedicados a desarrollar y mantener una solución propia (Miller y Pearson, 2011).

Otra variante es utilizar las aplicaciones que hay disponibles de código abierto. Existen múltiples herramientas especializadas que permiten desarrollar una solución que se adapte a los requerimientos que desea la empresa y las funcionalidades que demanda. Aunque en este caso se eliminen los costos en cuanto a licencias, requieren configurar y unir cada uno de los componentes generando la documentación, métodos y

procedimientos necesarios, lo que demanda tiempo y habilidades de los especialistas en el manejo y configuración de las aplicaciones.

Frente a los ataques y fallas de seguridad, los sistemas informáticos de seguridad siguen teniendo problemas en cuanto a la eficiencia de detección. Según el análisis del equipo de respuesta a incidentes de la compañía Verizon en 2012, del análisis de 855 incidentes se contabilizaron un total de 174 millones de registros comprometidos. En un 68 % del total, los activos comprometidos fueron servidores. Del total de incidentes, en el 75 % de los casos las fases de ataque e infiltración se completó en minutos, mientras que el 85 % tardó una o varias semanas en ser descubierto. Dentro de los datos que aporta el informe está que solo el 8 % de las grandes compañías (más de 1000 empleados) y un 1 % de resto, logró detectar la ocurrencia de los incidentes mediante el análisis de trazas y eventos de seguridad aun cuando en el 84 % de los casos estaba disponible la evidencia en las trazas (Baker et al., 2012).

Los datos mostrados, reflejan que las trazas generadas por dispositivos y aplicaciones siguen teniendo el papel fundamental en la prevención, detección y análisis de incidentes. Actualmente la mayoría de las empresas, independiente de su dimensión y las soluciones que tengan implantadas, no hacen uso en su totalidad de la información contenida en los registros generados o almacenados.

Aunque existen varias soluciones en el mercado, el incremento de su utilización no ha aportado una mejoría significativa respecto a datos de auditorías o después de la ocurrencia de incidentes de seguridad. Los sistemas de gestión de trazas son comúnmente subutilizados, generalmente se instalan para garantizar el cumplimiento de políticas y no se aprovechan los datos generados en el proceso de mejoras de los controles de seguridad(Dave Shackleford, 2008).

Generalmente la revisión de las trazas es una de las prioridades más bajas para los especialistas. Los departamentos de tecnología acostumbran a actuar ante la notificación de problemas, lo que hace que se accione sobre ellos una vez que ya han ocurrido. Sin un análisis de trazas periódico se limita la posibilidad de ser proactivo ante la ocurrencia de incidentes de seguridad. Según una encuesta anual realizada en por el Instituto SANS[1], la mayoría de las instituciones recolecta y almacena las trazas presentando la dificultad fundamental en extraer la información que pueda ser

[1] SANS(SysAdmin, Audit, Network, Security), http://www.sans.org/,

procesable. Para mejorar la automatización y análisis se requiere mejorar la integración y la correlación de la información entre los sistemas que permitan detectar ataques dentro del tráfico corriente(Jerry Shenk, 2014, 2012).

El proceso de gestión de trazas de seguridad requiere de variantes que integren la gran diversidad de formatos existentes. Es necesario garantizar la normalización de la información generada y su almacenamiento. También se requiere de herramientas que permitan buscar, agrupar y visualizar la información necesaria para la ejecución de tareas de seguridad.

Partiendo de los elementos planteados se identificó el siguiente **problema científico**: el proceso de gestión de trazas de seguridad requiere del análisis de grandes volúmenes de datos generados por múltiples fuentes lo que provoca lentitud en el procesamiento, búsqueda y visualización de la información contenida en las mismas.

El **objeto de estudio** que se trata es la gestión de trazas de seguridad y el **campo de acción** el procesamiento, búsqueda y visualización de trazas de seguridad.

Se plantea como **objetivo general** Desarrollar un marco de trabajo para la gestión centralizada de trazas de seguridad que contribuya a disminuir la demora en el procesamiento, búsqueda y visualización de la información.

Se plantean como **objetivos específicos**:

- Realizar un análisis de las principales regulaciones, normas y guías de buenas prácticas asociadas a la gestión de trazas.
- Diseñar un marco de trabajo para la gestión centralizada de trazas de seguridad utilizando herramientas de código abierto.
- Implementar un sistema centralizado de gestión de trazas de seguridad en el nodo de comunicaciones de la Universidad de las Ciencias Informáticas (UCI) aplicando el marco de trabajo propuesto.
- Validar el marco de trabajo propuesto a partir del sistema desplegado en el nodo de comunicaciones de la Universidad de las Ciencias Informáticas.

Como **hipótesis de investigación** se plantea: el desarrollo de un marco de trabajo para la gestión centralizada de trazas, que incluya los principales eventos de seguridad

y utilice herramientas de código abierto, facilitará el procesamiento de la información generada y contribuirá a disminuir el tiempo de búsqueda y visualización de los registros almacenados en las trazas.

En el desarrollo de la investigación se utilizaron los métodos teóricos hipotético-deductivo para la formulación de la hipótesis de trabajo y la concepción de los objetivos a alcanzar. También se empleó el método inductivo-deductivo para trabajar con los elementos extraídos de la bibliografía consultada elaborando conclusiones derivadas que ayuden a seleccionar, organizar y elaborar una propuesta sobre la base del conocimiento existente.

Como métodos empíricos se utilizaron el análisis documental en la revisión de la literatura especializada y la medición para la validación de los resultados obtenidos.

La **novedad** de la investigación consiste en el desarrollo de un marco de trabajo para la gestión centralizada de trazas de seguridad, que incluye la recolección de los eventos más importantes descritos en las principales regulaciones y normas. Como parte del desarrollo, se seleccionan las herramientas de código abierto para la recolección de estos eventos, siguiendo un enfoque integrador que extiende las propuestas existentes.

El **aporte teórico** de la investigación es el marco de trabajo propuesto.

Los **aportes prácticos** están dados por los elementos siguientes:

- Caracterización de un conjunto de herramientas de código abierto para la recolección de los principales eventos de seguridad.
- Descripción de una arquitectura de despliegue para la gestión centralizada de trazas de seguridad.
- Especificaciones y configuraciones necesarias para las herramientas de recolección y procesamiento de trazas.

El documento de tesis siguiente está estructurado en introducción, tres capítulos, conclusiones, recomendaciones, referencias bibliográficas y anexos. El contenido de los capítulos se distribuye de la siguiente forma:

- Capítulo 1: se presentan los conceptos esenciales asociados a la gestión de trazas así como el análisis de los protocolos y aplicaciones más utilizadas. Se detalla el funcionamiento del protocolo Syslog y se describen los principales sistemas de gestión centralizada de trazas de seguridad existentes.
- Capítulo 2: se propone un marco de trabajo para la gestión centralizada de trazas de seguridad. Se lleva a cabo la selección de las principales herramientas para obtener las fuentes de trazas que deben ser recolectadas acorde a las principales regulaciones, normas y guías de buenas prácticas. Se propone la arquitectura de despliegue correspondiente y cada uno de los componentes de software que la integran.
- Capítulo 3: en este capítulo se valida la implementación del marco de trabajo y la arquitectura propuesta en el entorno del nodo de comunicaciones de La universidad de las Ciencias Informáticas y se presentan los resultados obtenidos.

Capítulo 1. Gestión centralizada de trazas de seguridad. Herramientas disponibles

En el presente capítulo se muestran los conceptos esenciales asociados a la gestión de trazas de seguridad así como el análisis de los protocolos y aplicaciones más utilizadas. Se detalla el funcionamiento del protocolo Syslog y se describen los principales sistemas de gestión centralizada de trazas existentes.

1.1. Conceptos básicos

Un **registro de auditoría o traza** es una serie de registros de eventos generados sobre las actividades realizadas por el sistema operativo, las aplicaciones o los usuarios. El sistema operativo genera distintos tipos de trazas en dependencia del tipo o actividad(Guttman y Roback, 1995).

Desglosando cada uno de sus componentes: un **evento** es la ocurrencia simple en un entorno, usualmente asociado a un cambio de estado. Generalmente incluye un registro de tiempo y eventos que describen sobre sus causas o efectos. Los eventos se agrupan en **categorías de eventos** según cierta metodología u organización. Ejemplos pueden ser, según el tipo de dispositivo, impacto, etc.

Los eventos están compuestos por **campos.** Cada campo especifica una característica particular. Ejemplos de campos de eventos son la fecha, hora, Dirección IP o identificación de usuario.

Un **registro de evento** es una colección de campos que en su conjunto describen un evento. Sinónimos de registro de eventos son **registro de auditoría, traza y log**(Fitzgerald y Heinbockel, 2010). Muchas de las trazas generadas por los sistemas y las computadoras están asociadas a la seguridad o generan información relacionada. Este tipo de registro se conoce comúnmente como **traza de seguridad**. Ejemplo de trazas asociadas a la seguridad son los registros de acceso, las alertas de los sistemas detectores de intrusiones y cortafuegos. En el desarrollo de la investigación se

utilizarán los términos trazas y trazas de seguridad indistintamente, asumiendo el mismo nivel de importancia para el análisis y monitoreo de seguridad.

En una red, los sistemas, dispositivos y aplicaciones pueden llegar a generar gran cantidad de trazas. El análisis de trazas puede verse como un problema asociado al campo de *big data.* La información generada cumple con las características principales que son: la generación desde múltiples fuentes, diferentes tipos de datos y una estructura variable(Sawant y Shah, 2013).

1.2. Referencias a la gestión de trazas en estándares, guías y documentos de buenas prácticas de seguridad

Las trazas generadas por los sistemas de hardware y software tienen una importancia fundamental en el proceso de gestión de la seguridad de la información. Dentro de los principales documentos está la guía de buenas prácticas de la Organización Internacional para la Estandarización (ISO por sus siglas en inglés) y la Comisión Electrotécnica Internacional (IEC por sus siglas en inglés) ISO/IEC 27002 como parte de la norma certificable ISO/IEC 27001. La sección 12, seguridad de las operaciones, se dedica al tratamiento de las trazas y la subsección 12.4, registro y monitoreo, contiene cuatro controles asociados(ISO, 2005).

Estos son:

- Registro de eventos, con 15 tipos principales a registrar.
- Seguridad de las trazas almacenadas en cuanto a modificación o eliminación.
- Registro de las actividades de los administradores de sistemas.
- Sincronización de tiempo.

El estándar de Seguridad de Datos para la Industria de Tarjeta de Pago (PCI DSS por sus siglas en inglés) desarrollado por el "*Payment Card Industry Security Standards Council*" (PCI SSC) Propone 12 requerimientos a cumplir. El décimo requisito, con siete elementos a tener en cuenta asociados a: registro de acceso de forma individual, registro automático de eventos, campos que deben contener los eventos, sincronización de tiempo, seguridad de las trazas, revisión periódica y retención(Council, 2010).

La división de seguridad del Instituto Nacional de Estándares y Tecnologías de EEUU (NIST por sus siglas en inglés) desarrolla un conjunto de publicaciones especiales conocidas como Serie 800, ampliamente adoptadas por la comunidad de especialistas de seguridad. El documento SP 800-53, Controles de seguridad recomendados para los sistemas de información y organizaciones federales, dedica 14 controles específicos a la auditoría y gestión de trazas(NIST, 2007).

El estudio realizado por un grupo de expertos de seguridad, publicado bajo el título "*Twenty Critical Controls for Effective Cyber Defense: Consensus Audit Guidelines*" (CAG), describe 20 controles técnicos que son críticos e indispensables en un sistema de seguridad informática. El control 14: monitoreo, mantenimiento y análisis de las trazas de auditoría; propone 11 acciones concretas sobre el monitoreo y análisis de trazas(CSIS, 2013).

El grupo de seguridad de Comunicaciones electrónicas (CESG por sus siglas en inglés) del Reino Unido, publica una serie de documentos y guías de buenas prácticas de obligatorio cumplimiento para instituciones estatales. Compañías líderes en el área de la gestión de trazas y de eventos de seguridad, ofrecen productos que cumplen con la guía GPG13 que pertenece a esta serie. GPG13 consta de 12 aspectos para garantizar el monitoreo continuo de protección. La implementación de cada uno de los aspectos descritos está directamente relacionada con la capacidad de gestión de trazas(AccelOps, 2013).

En Cuba, de obligatorio cumplimiento para todas las entidades, está la resolución 127 de 2007 del Ministerio de la Informática y las Comunicaciones (MIC) actualmente Ministerio de las Comunicaciones (MICOM). La resolución cuenta con 100 artículos. La sección octava del Capítulo III trata sobre la seguridad de redes con tres artículos. Concretamente, los artículos 58 inciso b, 62 inciso b y 83 inciso b tratan sobre la generación, revisión periódica y específicamente, el registro de las conexiones remotas(MIC, 2007).

De las guías y documentos de buenas prácticas descritas se pueden extraer tres categorías asociadas a la gestión de trazas. Una categoría general asociada a los componentes necesarios en la implementación de un sistema de gestión de trazas, los registros de trazas de mayor importancia para la seguridad de los sistemas

monitorizados y a un nivel más profundo, los campos que deben contener estos registros. La agrupación de las categorías se muestra en la Figura 1.1.

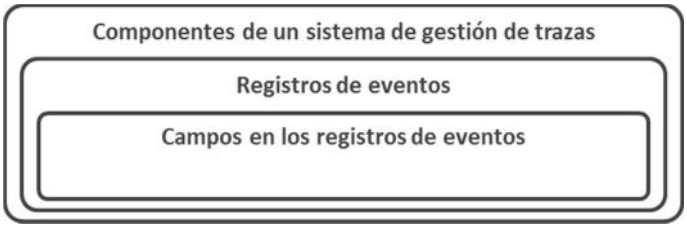


Figura 1.1. Categorías asociadas a la gestión de trazas

Por otra parte, se puede destacar que la ISO/IEC 27002 hace mayor énfasis en los registros de eventos. PCI DSS propone explícitamente los campos que deben contener los registros de eventos. El resto está orientado principalmente en los componentes que debe conformar un sistema de gestión de trazas.

1.2.1. Guía para la gestión de trazas de seguridad (NIST SP800-92)

El documento del NIST SP800-92(Kent y Souppaya, 2006) está dedicado completamente la gestión de trazas. En cinco secciones, trata las necesidades y retos de las organizaciones, arquitectura y funciones así como las recomendaciones para la elaboración de políticas y los procesos a llevar a cabo en las operaciones de gestión de trazas.

SP800-92 propone una infraestructura de gestión de trazas compuesta por tres capas:

- **Generación de trazas**: en esta capa, van a estar los servidores, estaciones de trabajo y dispositivos que generan los datos.
- **Análisis y almacenamiento de trazas**: esta capa está compuesta por uno o más servidores de trazas que reciben los datos, ya sea en tiempo real o mediante aplicaciones que envían las trazas cada cierto tiempo. Las trazas son almacenadas en el propio servidor o en servidores de almacenamiento dedicado.
- **Monitoreo de trazas**: contiene las consolas para la revisión, análisis automático y generación de reportes.

La guía a su vez plantea la implementación en tres etapas como se muestra en la Figura 1.2.

En el proceso de planeación, la definición de políticas efectivas para la gestión de trazas, requiere que se determinen con claridad las áreas de responsabilidad para los distintos grupos asociados a las redes, los sistemas, la seguridad y el resto del personal en la entidad.

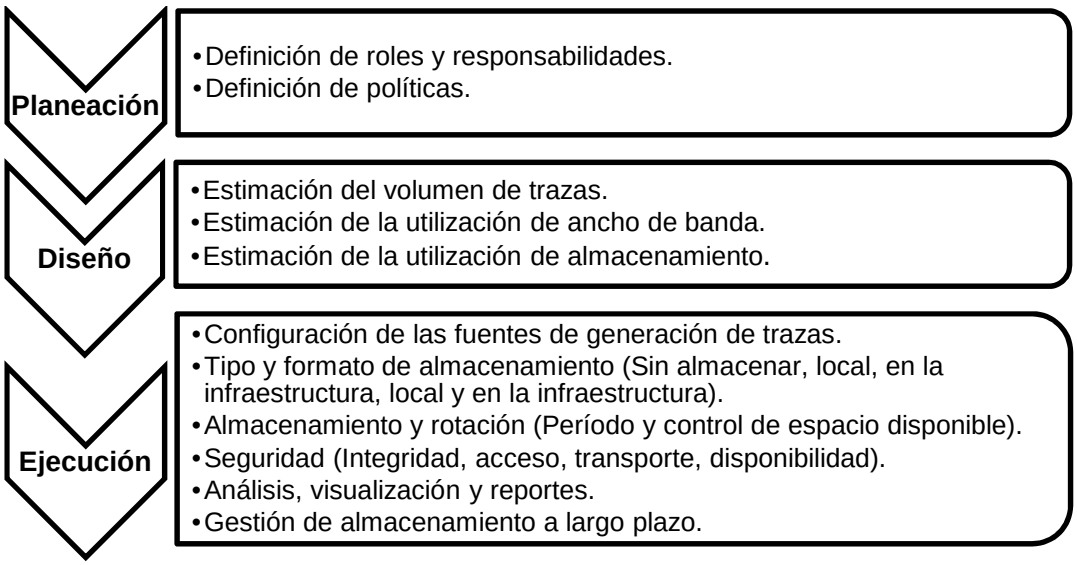


Figura 1.2. Etapas de implementación de un sistema de gestión centralizado de trazas siguiendo la guía del NIST SP800-92.

La definición de roles, asociada al personal que está involucrado directamente, puede agruparse en las categorías que se muestran:

- Personal del grupo de administración de infraestructura y sistemas.
- Administradores de seguridad.
- Desarrolladores de aplicaciones.
- Grupo de seguridad informática.
- Grupo de respuesta a incidentes.
- Auditores.
- Jefes de áreas.

En dependencia del tamaño y organización de la entidad, varios de los roles descritos pueden agruparse en uno genérico.

Acorde con la guía NIST SP800-92 se debe definir políticas para cada una de las etapas del proceso de gestión de trazas. Las políticas deben tener como objetivo la reducción de riesgos buscando un equilibrio con el tiempo y recursos necesarios para su implementación.

El documento SP 800-92 contiene los pasos necesarios para el desarrollo de una infraestructura de gestión de trazas. Describe la implementación de la gestión centralizada de trazas utilizando Syslog como protocolo. Con Syslog todas las fuentes generadoras tienen el mismo formato. Es ampliamente utilizado de forma nativa. Existen un gran número de aplicaciones que lo implementan o proveen mecanismos para la conversión de las trazas desde otros formatos. El protocolo Syslog será tratado en detalle en el desarrollo del capítulo.

Otra variante es la utilización de sistemas de gestión de eventos de información de seguridad (SIEM). Los sistemas SIEM incluyen la gestión centralizada de eventos de seguridad. Tienen la ventaja de poder manejar múltiples formatos de trazas. Permiten tareas de reducción de eventos, correlación, priorización de eventos de seguridad significativos y mostrar los resultados en interfaces de visualización. Además tienen funcionalidades como la creación de bases de conocimientos y la gestión de incidentes.

En cuanto a la utilización de los SIEM para la gestión de trazas, Anton Chvakin(Anton Chuvakin, 2010) hace un análisis comparativo donde expone las principales diferencias en cuanto a la función de ambos sistemas así como las principales dificultades a la hora de seleccionar la tecnología adecuada. Plantea que la gestión de trazas no puede limitarse a la recolección de eventos asociados a incidentes de seguridad solamente. Chuvakin se refiere a que se debe garantizar primeramente la gestión de trazas, apuntando que no se puede aumentar la velocidad de respuesta ante un incidente si no se mejora primero la respuesta que se brinda ante la ocurrencia.

Un análisis detallado de la estructura y funcionamiento de los sistemas SIEM puede consultarse en el libro: “Security Information and Event management” de Henry Miller (Miller y Pearson, 2011).

Aunque la guía NIST SP800-92 cubre los procesos clásicos de la gestión de trazas, las redes, los sistemas y la seguridad han ido evolucionando aceleradamente. Nuevos

actores han tomado protagonismo; como la computación en la nube, la gestión de grandes volúmenes de datos y la seguridad como servicio. El documento analizado no ha sido objeto de una actualización desde su publicación en 2006. Como un documento generalizador, no plantea una arquitectura de despliegue donde se integren cada uno de los componentes en una solución aplicada. De igual forma, trata las fuentes de generación de trazas haciendo una clasificación global, no propone una selección de eventos según su importancia, resumiendo solamente una lista genérica de software de seguridad así como la información que debe ser tomada en cuenta en cuanto a los sistemas operativos y aplicaciones de usuario.

Otro elemento importante, es el análisis de las herramientas y recursos propuestos en el apéndice C del documento; nuevos formatos de trazas y aplicaciones de software que no se mencionan, han sido desarrollados y ocupan un rol importante en el proceso de gestión de trazas de seguridad.

1.3. Fuentes de generación de trazas

La generación de trazas se divide en cuatro grandes grupos (Anton Chuvakin, 2010).

- Un primer grupo denominado trazas de seguridad, cuyo objetivo es detectar y responder ante ataques, robo de información u otro evento que pueda comprometer la confidencialidad, integridad o disponibilidad de la información y los sistemas.
- El próximo grupo está asociado a las trazas de operación. La información generada informa a los operadores sobre fallos o posibles acciones que deben ser ejecutadas. Dentro de este grupo, los registros más comunes son las trazas de acceso a sitios web que pueden ser utilizadas para aprovisionamiento, trazas de auditoría u otras aplicaciones.
- Como tercer grupo están las trazas generadas para el cumplimiento de regulaciones. Este tipo de trazas generalmente incluye las asociadas a seguridad y servicios dado que uno de los objetivos es mejorar la seguridad en los sistemas.
- Por último está el registro de trazas de depuración. Son utilizadas fundamentalmente por desarrolladores y no se activan en sistemas en producción por el efecto negativo que tiene sobre el rendimiento.

Las fuentes de generación de trazas pueden agruparse siguiendo criterios asociados al tipo de aplicaciones o a su funcionalidad.

Según el tipo de aplicaciones, se tienen trazas de software asociado directamente a la seguridad y trazas relativas a los sistemas operativos, las aplicaciones y servicios que se encuentren en ejecución(Kent y Souppaya, 2006). Los registros asociados a las categorías mencionadas se muestran en la Tabla 1.1.

Una clasificación más genérica, sería la agrupación de las trazas asociadas a la seguridad relativas a los *hosts* y las trazas generadas por dispositivos y aplicaciones de red(Chuvakin, Schmidt y Phillips, 2012).

Tabla 1.1. Agrupación lógica de las fuentes de generación de trazas según el tipo de aplicaciones(Kent y Souppaya, 2006).

Software de seguridad	Sistema operativo	Aplicaciones
Software antimalware	Eventos del sistema	Peticiones en aplicaciones tipo cliente servidor
Sistemas de detección y prevención de intrusiones	Registros de auditoría	Información de auditoría
Software de acceso remoto		Información de utilización de las aplicaciones
Proxies web		Trazas de funcionamiento y operación significativas
Software de gestión de vulnerabilidades		
Servidores de autenticación		
Routers		
Firewalls		
Servidores de red en cuarentena		

Chris Fry, expone que la selección de las fuentes de eventos para el monitoreo de seguridad, debe realizarse partiendo de la premisa de qué tipo de utilización se le dará a la información recolectada, ya sea para el monitoreo, respuesta a incidentes o cumplimiento de regulaciones y análisis forense. Cada tipo tiene su característica distintiva como se muestra en la Tabla 1.2 (Fry y Nystrom, 2009).

Tabla 1.2. Agrupación de las fuentes de eventos y sus características.

Selección de eventos	Características
Monitoreo	Requiere flujos continuos de eventos lo más cercano posible a tiempo real. No requiere de grandes espacios de almacenamiento.
Respuesta a incidentes e investigaciones	Requiere de espacio necesario para la ejecución de búsquedas rápidas de los eventos más recientes. El tiempo de retención va a depender del espacio y volumen del flujo de eventos.
Cumplimiento de regulaciones, normas y análisis forense	Requiere del almacenamiento a largo plazo, generalmente años. El acceso a los datos en dependencia del almacenamiento, puede demorar horas o días.

De cada grupo se deben recolectar aquellas trazas que estén en concordancia con las políticas definidas. Garantizar el almacenamiento lo más cercano posible a tiempo real, para poder realizar operaciones de monitoreo y tener en cuenta el almacenamiento a mediano y largo plazo en la respuesta a incidentes y cumplimento de regulaciones.

1.4. Componentes principales y eventos que deben ser registrados en un sistema de gestión de trazas

Tomando como base una selección de los principales documentos existentes, en cuanto a la gestión de seguridad y buenas prácticas, se muestra en la Tabla 1.3 una síntesis de los componentes asociados a la gestión de trazas que se tienen en cuenta por cada documento analizado.

Para medir la importancia de cada recomendación, se revisó su aparición, obteniendo como resultado cuáles tenían mayor presencia y por tanto serían los más relevantes a tener en cuenta.

Tabla 1.3. Síntesis de los principales componentes por cada una de las guías de buenas prácticas y estándares fundamentales en cuanto a la gestión de trazas.

Componentes	27002	PCIDSS	800-53	CAG	GPG13	MIC 127
Generación y registro de trazas	X	x	x	x	x	x
Sincronización de tiempo	X	x	x	x	x	-
Seguridad de las trazas	X	x	x	x	-	-

Revisión periódica de las trazas	X	x	x	x	x	x
Retención	-	x	x	x	-	x
Políticas y procedimientos de auditoría	-	-	x	-	-	-
Definir eventos auditables	-	-	x	-	-	-
Contenido de los registros de eventos	X	x	x	x	-	-
Capacidad de almacenamiento	-	-	x	x	-	-
Respuesta a fallos del sistema de auditoría	-	-	x	-	x	-
Reducción de trazas	-	-	x	x	-	-
No repudio	-	-	x	x	-	-

En la Figura 1.3 se muestra el resultado de la presencia de cada componente y como la generación, revisión periódica, tener una fuente de tiempo confiable, qué información deben tener las trazas que se almacenan, por cuanto tiempo se van a conservar y su seguridad; deben ser los aspectos primarios a considerar.

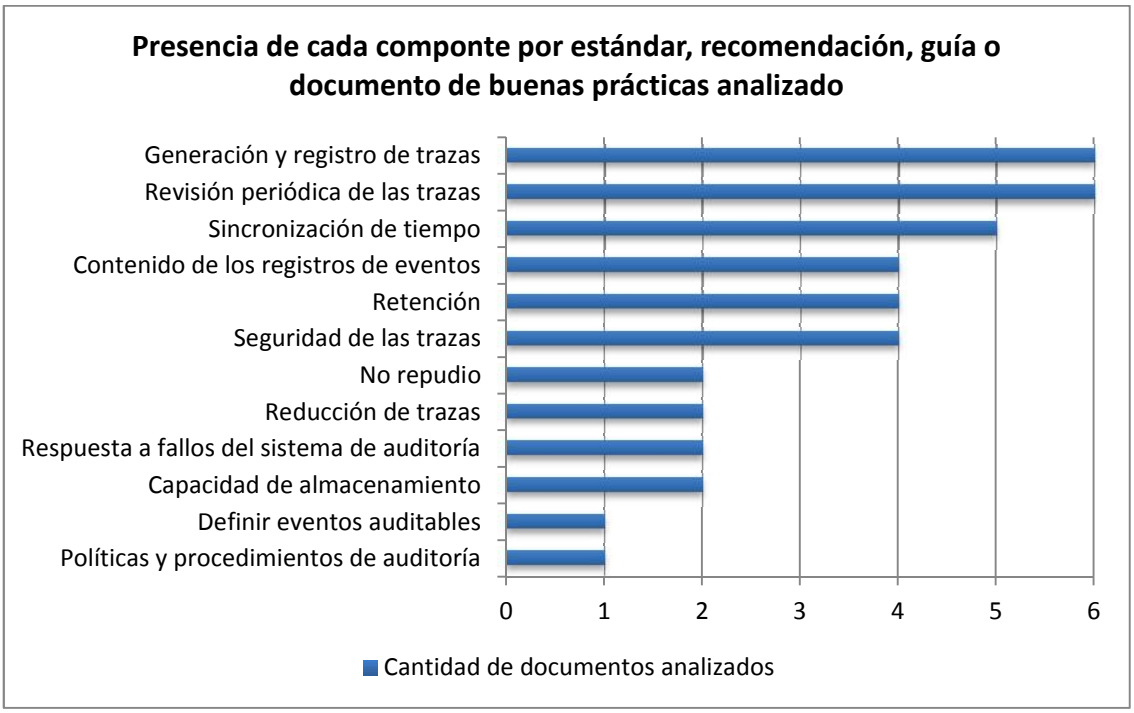


Figura 1.3. Presencia de cada componente por guía o norma de buenas prácticas de seguridad analizada.

En la Tabla 1.4 se muestra una síntesis de la información que debe ser registrada y almacenada para su análisis y presentación de reportes. Se agrupan en nueve tipos de registros de eventos genéricos, presentes explícitamente al menos en uno de los documentos analizados.

Cada registro de evento idealmente debe proveer un conjunto de información que permita obtener una trazabilidad, lo más completa posible, del evento ocurrido. Para evaluar la calidad del registro deben intentar responderse cinco preguntas que se conocen por su terminología en inglés como las cinco W's(Chuvakin, Schmidt y Phillips, 2012).

1. ¿Qué pasó?
2. ¿Cuándo pasó?
3. ¿Dónde pasó?
4. ¿Quién está involucrado?
5. ¿Cuál fue la fuente de origen?

Tabla 1.4. Principales registros de eventos y su presencia en cada una de las principales guías y estándares.

Registro de evento	**Mapeo contra las principales regulaciones y guías de buenas prácticas**
Acciones con privilegios administrativos	27002; PCIDSS; CAG
Acceso a las trazas	PCIDSS
Monitoreo de sesiones	NIST 800-53, GPG13
Eventos satisfactorios y fallidos de usuarios, aplicaciones y sistemas	PCIDSS;800-53;CAG;GPG13;27002
Creación, modificación y borrado de objetos del sistema	PCIDSS;27002
Monitoreo de fuga de información	NIST 800-53;GPG13
Registro de conexiones remotas	CAG;GPG13;127/2007
Registro de trazas tráfico de red	27002;GPG13
Eventos críticos	GPG13

Dentro de las guías analizadas, PCI DSS detalla explícitamente qué campos deben tener los registros de eventos. En la ISO/IEC 27002, junto con los eventos que se

deben registrar, se mencionan además algunos de los campos que deben contener las trazas. La guía del NIST 800-53 dentro de la familia de controles, auditoria y contabilidad, incluye el control “AU-3” contenido de los registros de eventos. La Tabla 1.5 muestra un resumen de los eventos propuestos en estas guías y su relación con la información necesaria para garantizar la trazabilidad de un evento.

Tabla 1.5. Asociación de los campos de registros de eventos a los que se hace referencia explícitamente en las guías ISO/IEC 2002, PCI DSS y SP 800-53 junto con la información que proveen.

Campos	Responde a	PCI DSS	ISO/IEC 27002	SP 800-53
Tipo de eventos. Satisfactorios y fallidos	Qué pasó	x	-	x
Fecha y hora	Cuándo pasó	x	x	x
Nombre del objeto afectado	Dónde pasó	x	-	x
Direcciones de red y protocolos	Quién está involucrado	-	x	x
Fuente donde se generan los eventos	Cuál fue la fuente de origen	x	x	x

Si bien la lista de la Tabla 1.3 responde idealmente a las cinco preguntas propuestas, esta tarea recae principalmente en los desarrolladores de software, que muchas veces no la tienen en cuenta. El resultado obtenido, es una gran diversidad de los mensajes en las trazas que no siempre pueden responder a las preguntas mencionadas.

1.5. Diversidad de formatos de trazas

La mayoría de las trazas actualmente no siguen ningún formato predeterminado. Solo tienen en común, en algún campo del mensaje, la fecha y hora. Las trazas se pueden generar en ficheros binarios o ficheros de texto plano. Los formatos utilizados en la generación pueden estar asociados a algún documento *“Request for Comment”* (RFC), o pertenecer a un formato propietario.

Dentro de los ficheros binarios están los generados por la gama de los sistemas operativos Windows. El manejo de eventos en Windows se reestructuró a partir de Windows 7 cambiando la estructura, formato y mecanismo utilizado para el

almacenamiento. El sistema de registro de eventos se conoce como *"Windows Event Logs".* Las trazas generadas son almacenadas en formato XML[2] binario y se organizan en dos grupos:

- Trazas de Windows. Incluyen Aplicaciones, Sistema, Seguridad, Inicio del sistema y Eventos reenviados.
- Trazas de aplicaciones y servicios. Incluyen la generación de trazas de aplicaciones específicas y otros eventos del sistema.

Los eventos son almacenados según la configuración predeterminada del sistema en: *C:\Windows\system32\winevt\Logs*(Carvey, 2014)

Se registran cinco tipos de eventos como se muestra en la Tabla 1.6.

Tabla 1.6. Tipos de eventos para sistemas operativos de Microsoft. Windows Vista y versiones superiores(Microsoft, 2015b).

Tipos de eventos	Descripción
Error	Eventos sobre un problema significativo como la pérdida de datos o de una funcionalidad.
Alerta	Eventos que no necesariamente son significativos pero indican la posible ocurrencia de un error futuro.
Información	Evento que describe la ejecución satisfactoria de una aplicación, dispositivo o servicio.
Auditoría satisfactoria	Registro de auditoría asociado a un acceso satisfactorio.
Auditoría fallida	Registro de auditoría asociado a un acceso fallido.

Los eventos se componen según un conjunto de campos definidos por el sistema mediante la estructura *EVENTLOGRECORD*. Uno de los campos más importantes de la estructura es *eventID,* que funciona de identificador para una acción determinada(Microsoft, 2015a). Ejemplos de eventos pueden ser el 4624 que indica que una cuenta se ha autenticado satisfactoriamente o el 4634 indicando que una

[2] "***Ex****tensible* ***M****arkup* ***L****anguage"*(XML). Lenguaje extensible de marcas desarrollado por el Consorcio World Wide Web (W3C) utilizado para la descripción de datos. http://www.w3c.es/Divulgacion/GuiasBreves/TecnologiasXML

cuenta ha cerrado la sesión. Los eventos de Windows y su descripción pueden consultarse en línea en el sitio web de Randy Franklin Smith, *Windows Security Log Encyclopedia*[3]. Cada evento generado tiene un tamaño aproximado de 500 bytes dentro del fichero binario(Microsoft, 2009).

Los eventos generados pueden ser consultados directamente mediante el visor de eventos de Windows como se muestra en la Figura 1.4.

Con formato binario también se pueden encontrar los ficheros *utmp* y *wtmp* generados en los sistemas GNU/Linux y UNIX; que permiten tener un registro de auditoría de accesos de inicio y cierre de sesión así como quién se encuentra autenticado en cada momento en el sistema(Michael Kerrisk, 2014). Los ficheros comprimidos son otra variante que también se clasifican como binarios.

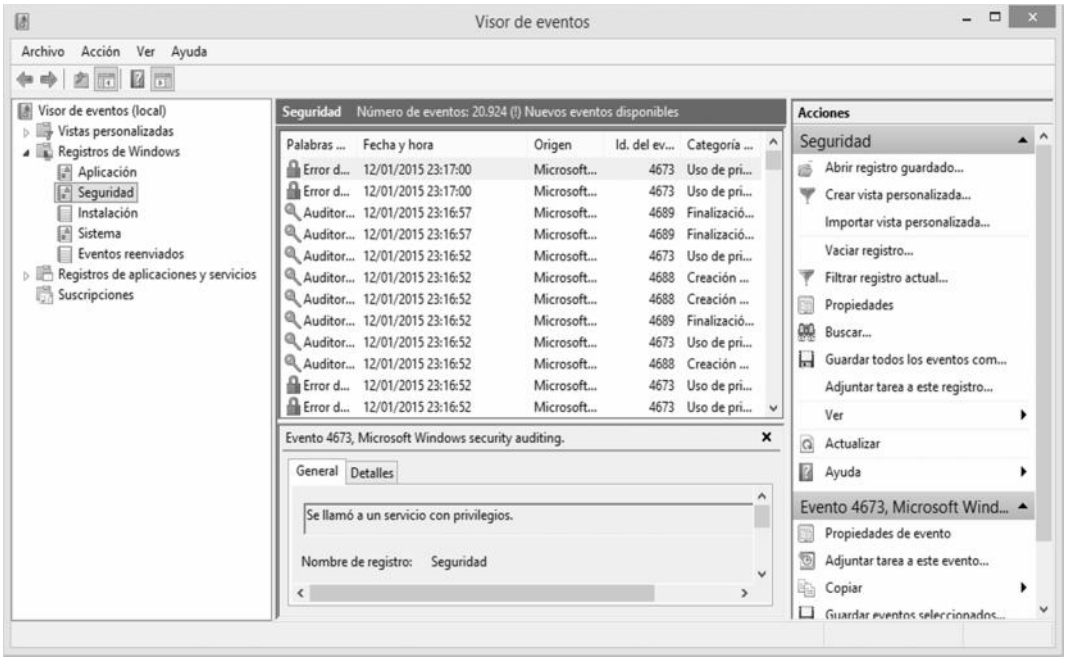


Figura 1.4. Visor de eventos de Windows 8.1.

Para la generación de ficheros de trazas en texto plano el formato más común está definido mediante la utilización del protocolo Syslog que se describe en el siguiente epígrafe.

[3]Windows Security Logs Events.
https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/Default.aspx

1.5.1. El protocolo Syslog

El protocolo Syslog es un estándar de facto para el envío y recepción de mensajes entre los sistemas operativos, documentado inicialmente según la RFC3164(C. Lonvick, 2001). El protocolo original contenía múltiples deficiencias, siendo revisado en 2009 cuando se define como estándar formalmente según la RFC5424(R. Gerhards, 2009).

El protocolo se basa en tres capas:

- **Capa de contenido**: maneja la información de gestión contenida en los mensajes.
- **Capa de aplicación**: maneja la creación, interpretación, encaminamiento y almacenamiento de los mensajes.
- **Capa de transporte**: se encarga de situar y recolectar los mensajes de la red.

Como se muestra en la Figura 1.5, en la capa de contenido se van a generar los eventos que se transforman al formato Syslog. La capa de aplicación se encarga de recolectar los mensajes recibidos de la capa de transporte y realiza la función de retransmisión, volviendo a enviar los mensajes recolectados o generados por el propio sistema. La capa de transporte se encarga de enviar y recibir los mensajes.

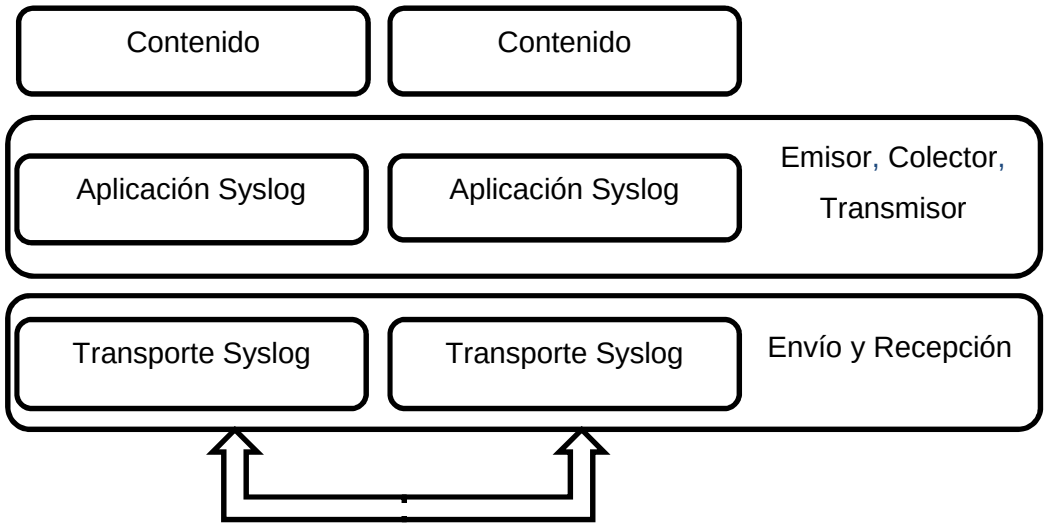


Figura 1.5. Estructura en capas del protocolo Syslog(R. Gerhards, 2009).

Syslog se basa en un conjunto de principios básicos para funcionar, como un protocolo simple sin mecanismos de chequeo en la recepción de mensajes:

- No provee ningún mecanismo para recibir confirmaciones de envío.
- Los iniciadores de mensajes y de retransmisión pueden ser configurados para enviar mensajes hacia múltiples colectores u otro sistema de retransmisión.
- Los iniciadores de mensajes, retransmisión y colectores pueden residir en el mismo sistema.

Syslog posibilita, siguiendo los principios enunciados, conformar múltiples arquitecturas de despliegue. Un nodo iniciador puede conectarse directamente a un nodo colector o enviar los mensajes a varios nodos retransmisores en cascada o en paralelo, que a su vez pueden reenviar los mensajes a uno o varios nodos colectores. Varios de los posibles esquemas se muestran en la Figura 1.6 y pueden consultarse en más detalle en el epígrafe 4.1 de la RFC5424.

La RFC3164 describe la utilización del protocolo UDP[4] para el envío de mensajes Syslog asociado por defecto el puerto 514. Muchos sistemas siguen esta implementación para el envío de mensajes. UDP restringe el tamaño de los mensajes en muchas implementaciones a un kilobyte. La especificación de la RFC3164 contiene un conjunto de deficiencias asociadas a la no fiabilidad del protocolo UDP. Los mensajes pueden perderse o ser descartados, es muy limitado el tamaño del mensaje y no existe un mecanismo de seguridad que prevenga los posibles ataques de modificación a los mensajes.

La definición de Syslog según la RFC 5424 no especifica ningún protocolo de transporte, describe la estructura de los mensajes de forma independiente al tipo de transporte, lo que permite utilizar mecanismos más fiables en la trasmisión. La RFC5426 define UDP para el transporte, desarrollado fundamentalmente para mantener la compatibilidad con la implementación original. La RFC3195 utiliza TCP como protocolo fiable, la RFC5425 describe la utilización de TLS para asegurar la transmisión y la RFC5848 establece mecanismos para garantizar la integridad de los mensajes, autenticación de origen, resistencia a los ataques de reinyección y detección de mensajes perdidos(F. Miao, 2009; A. Okmianski, 2009; M. Rose, 2001; J. Kelsey, 2010). Las implementaciones actuales de las herramientas para el manejo de mensajes utilizando Syslog mantienen el límite del tamaño de los mensajes a un kilobyte solo por compatibilidad y puede ser modificado desde la configuración de la herramienta.

[4] User Datagram Protocol(UDP). Protocolo no fiable de nivel de transporte.

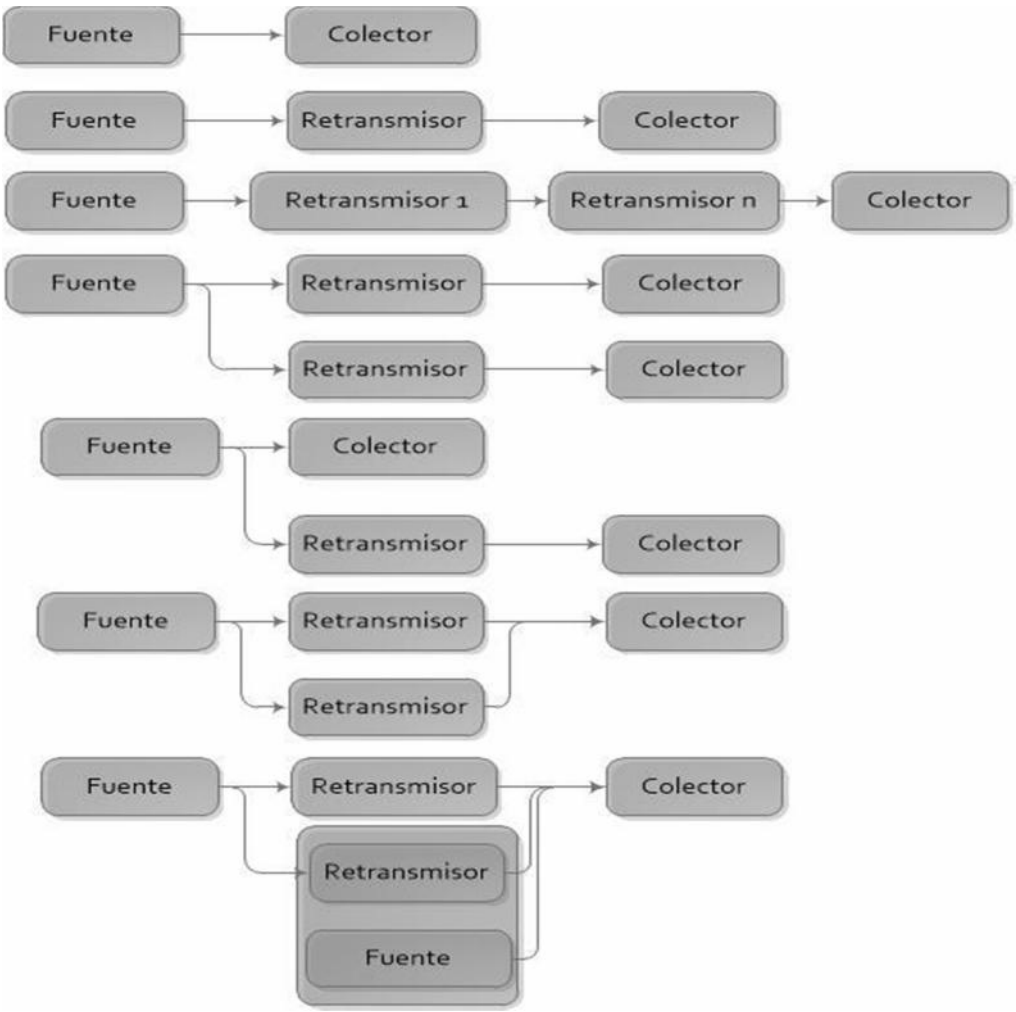


Figura 1.6. Mecanismos de transporte(R. Gerhards, 2009).

Aunque TCP como protocolo de transporte aumenta la fiabilidad en el envío y recepción, no está excepto de problemas. Rainer Gerhards, desarrollador de una de las implementaciones de Syslog más utilizadas en las distribuciones GNU/Linux y autor en varias RFC asociadas a los mecanismos de registros de trazas, expone las limitaciones en cuanto a fiabilidad de TCP y la necesidad de que exista confirmación a nivel de aplicación para garantizar la fiabilidad en la comunicación(Rainer Gerhards, 2008) Propone como solución el protocolo *Reliable event logging protocol* (RELP) implementado sobre TCP y que añade una capa de confirmación de envío y recepción a nivel de aplicación(Rainer Gerhards, 2014a). Källqvist y Lan exponen un análisis del protocolo RELP en una implementación de gestión de trazas centralizada en un

entorno de alta disponibilidad. Comparan la implementación con la versión comercial de Syslog-NG que incluye un protocolo similar(KÄLLQVIST y LAM, 2012).

Syslog tiene como característica más significativa, que su inclusión o soporte por casi la totalidad de las aplicaciones y sistemas existentes para la generación de trazas y fundamentalmente su diseño de agentes de generación, retransmisión, y recolección; facilita su utilización para centralizar las trazas desde diversas fuentes.

El formato de mensajes Syslog (ver Anexo 1), según la notación argumentada de Backus-Naur **(**ANBF por sus siglas en inglés) está compuesto por un encabezado, un campo de datos estructurados opcional y el campo mensaje.

SYSLOG-MSG = HEADER SP STRUCTURED-DATA [SP MSG]

El campo HEADER lo componen:

- La prioridad.
- El número de versión del protocolo.
- La fecha y hora en que se origina el mensaje.
- El nombre de la aplicación que genera el mensaje.
- El nombre del dispositivo que origina el mensaje.
- Identificador del proceso asociado a la aplicación y un identificador de mensaje.

El campo prioridad es el resultado de la combinación de un valor numérico asignado a una lista de las aplicaciones más comunes denominadas *facility* (ver Anexo 2) y un valor denominado *severity* (ver Anexo 3) que prioriza la importancia del mensaje. Para calcular la prioridad se multiplica el valor asociado a la aplicación por ocho y se le suma el valor de importancia (severity*)* que contenga.

El campo STRUCTURED-DATA se utiliza para brindar información en un formato bien definido y que sea fácil de interpretar por otras aplicaciones. Puede contener información específica asociada al propio protocolo o a la aplicación que genera el mensaje.

El campo MSG contiene un texto sin ninguna especificación de formato, en una única línea, que brinda información sobre el evento generado. Si el mensaje comienza con el carácter especial denominado Marca de Orden de bytes (BOM por sus siglas en

inglés), indica el orden de los bytes y la utilización de caracteres Unicode, en este caso, para UTF-8[5].

El tamaño del mensaje lo define el tipo de transporte que se use sin que haya una especificación límite. Como requisito se debe ser capaz de manejar mensajes con longitud de 2048 octetos de byte.

Un ejemplo de mensaje en formato Syslog, documentado por Lonvick en la RFC 3164, según las estructuras descritas anteriormente, puede ser:
<34>1 2003-10-11T22:14:15.003Z mymachine.example.com su - ID47
- BOM'su root' failed for lonvick on /dev/pts/8
En este ejemplo la prioridad es 34, el mensaje fue creado el 11 de octubre de 2003 a las 22:14:15.003 pm. Se originó en la máquina *mymachine.example.com* la aplicación que lo generó fue *"su",* no se conoce el identificador del proceso (-), el id del mensaje es ID47 el mensaje no contiene el campo STRUCTURED-DATA, y el mensaje contiene el texto: *'su root' failed for lonvick on /dev/pts/8* codificado en UTF8.

Muchas aplicaciones actualmente no soportan la RFC5424 y utilizan la descripción dada en la RFC3164. Las trazas estructuradas según las especificaciones Syslog, aunque presentan parámetros preestablecidos en el encabezado, la mayor cantidad de información se encuentra en el campo MSG dificultando el procesamiento de los mensajes.

Almacenando la información en ficheros de texto, se pueden utilizar herramientas de búsqueda de patrones, aunque su configuración se complejiza cuando se requiere de búsquedas sobre grandes cantidades de datos.

Las trazas en formato binario tienen como ventajas que requieren de menos espacio de almacenamiento y recursos del sistema para su generación. Por otro lado no pueden ser revisadas sin la utilización de una herramienta específica que permita interpretar y mostrar la información almacenada.

[5]UTF-8. Formato de codificación de caracteres estándar para la representación, presentación y procesamiento de texto http://www.unicode.org/faq/utf_bom.html

1.6. Otros estándares de formatos de mensajes

La falta de un estándar ha traído como consecuencia que cada desarrollador defina el suyo propio, por lo que existen prácticamente tantos formatos para la transmisión de trazas como aplicaciones desarrolladas.

Algunos de los formatos de trazas son:

- Common Log Format (CLF). Formato de trazas comúnmente utilizado para registrar los accesos en servidores web[6].
- ArcSight Common Event Format (CEF): formato propietario para la gestión de eventos utilizado en el sistema SIEM ArcSight[7].
- Cisco Intrusion Detection Event Exchange (SDEE/CIDEE): formato para el intercambio de mensajes entre dispositivos de seguridad para productos CISCO[8].
- Intrusion Detection Message Exchange Format (IDMEF): define mediante la RFC4765 el formato de datos y procedimientos de intercambio para el intercambio de mensajes entre los sistemas de detección de intrusos. Utiliza XML para la estructura de los mensajes[9].

Muchos de estos formatos, tienen como característica fundamental que fueron desarrollados para aplicaciones específicas por lo que se hace difícil su integración con otras soluciones de software.

Tabla 1.7. Resumen de las principales características de los distintos formatos de trazas(Chuvakin, Schmidt y Phillips, 2012).

	Trazas en XML	**Trazas en formato Syslog**	**Trazas en formato Texto**	**Trazas en formato propietario**
Consumo	Mediante la utilización de	Mayormente lectura	Solo lectura manual	Solamente mediante la

[6] Common Log Format (CLF). https://tools.ietf.org/html/rfc6872
[7] ArcSight Common Event Format (CEF) https://protect724.hp.com/servlet/JiveServlet/downloadBody/1072-102-6-4697/CommonEventFormat.pdf
[8] Cisco Intrusion Detection Event Exchange http://www.cisco.com/c/en/us/td/docs/security/ips/specs/CIDEE_Specification.html
[9] The Intrusion Detection Message Exchange Format (IDMEF) http://ietf.org/html/rfc4765

	aplicaciones.	manual.		utilización de aplicaciones
Comúnmente utilizadas	Trazas de seguridad	Trazas de operación, trazas de depuración.	Trazas de operación, trazas de depuración	Registro de trazas para alto desempeño
Ejemplo	Dispositivos CISCO detectores de intrusos	La mayoría de los ruters y switches	La mayoría de las aplicaciones para depuración	Checkpoint Firewall
Recomendadas	Transferir gran cantidad de información estructurada	Utilizado en la mayoría de los procesos de operación.	Añadir información estructurada para simplificar el análisis	Solamente para proceso de alto desempeño
Desventajas	Bajo desempeño, mensajes de gran tamaño	Inexistencia de estructura lo que dificulta el análisis	Generalmente las trazas solamente son comprendidas por los desarrolladores	Imposible leer directamente sin La utilización de una aplicación que convierta de formato binario a texto

En la Tabla 1.7 se muestra un resumen de las principales características de los diferentes formatos de traza. La diversidad existente es una de las dificultades principales en el proceso de normalización y análisis. Es una de las principales necesidades, contar con una estructura normalizada y que sea adoptada por los sistemas y aplicaciones. Un esfuerzo para lograrlo se comenzó con la especificación CEE.

1.7. Expresión común de eventos

Expresión común de eventos(CEE por sus siglas en inglés)(The MITRE Corporation, 2010) surge como una iniciativa del MITRE[10] en colaboración con varias industrias del

[10] La corporación MITRE es una organización sin ánimos de lucro que provee soporte en cuanto a tecnologías de la información al gobierno de los Estados Unidos. http://www.mitre.org/

área tecnológica de los Estados Unidos con el objetivo de estandarizar el proceso de describir, representar e intercambiar registros de eventos entre los sistemas electrónicos.

Tiene como objetivo mejorar los procesos de auditoría facilitando a los usuarios la interpretación de los registros de trazas, a la vez que posibilita la creación de registros útiles por las aplicaciones.

CEE Presenta una arquitectura de cuatro componentes representados en la Tabla 1.8 que se combinan para procesar un evento hasta convertirlo en un registro de traza.

Tabla 1.8 Componentes de la arquitectura CEE.

Componente	Descripción
CDET	Diccionario y taxonomía
CLS	Sintaxis de trazas
CLT	Transporte de trazas
CELR	Recomendaciones para los registros de eventos

Cuando un evento se genera, se utilizan como guía las recomendaciones definidas en CELR. Partiendo de la taxonomía y los diccionarios CEDT se seleccionan los campos que se van a registrar. Para generar el registro de los eventos se utiliza el lenguaje CLS y finalmente según el protocolo definido por el componente CLT se puede compartir el registro o transmitirlo a un repositorio de trazas. La arquitectura CEE tiene la particularidad de ser bidireccional pudiéndose recrear los eventos a partir de las trazas.

El desarrollo de CEE como estándar se detuvo según una nota publicada en su sitio oficial[11], aunque como un proyecto alternativo, se creó Lumberjack[12] para implementar las especificaciones descritas. La implementación utiliza los mensajes tradicionales Syslog encapsulando el campo MSG en formato JSON siguiendo las especificaciones CEE y añadiendo una etiqueta ("@cee:") al inicio de los datos JSON. Además del proyecto mencionado, otros desarrolladores de aplicaciones de gestión de trazas también adoptaron esta estructura

[11] CEE. https://cee.mitre.org/
[12] https://fedorahosted.org/lumberjack/

Aunque el desarrollo de un estándar para la gestión de trazas no se continuó, posibilitó la implementación de soluciones de software que manejan el formato Syslog con mensajes estructurados manteniendo la compatibilidad con la definición existente en la RFC5424 y posibilitando la integración con sistemas de indexado para la búsqueda y análisis en grandes volúmenes de trazas.

1.8. Arquitectura de gestión centralizada de trazas

La gestión centralizada de trazas consiste en el almacenamiento centralizado de los registros generados por los dispositivos de hardware y aplicaciones instaladas. El protocolo Syslog ha sido la base más utilizada en múltiples implementaciones proveyendo un marco simple para la generación, transmisión y almacenamiento de trazas. Múltiples aplicaciones tanto propietarias como de código abierto toman a Syslog como base añadiendo otras funcionalidades.

Dentro de la gestión centralizada de trazas se pueden agrupar las aplicaciones que no incluyen ningún sistema para la visualización como es el caso de la implementación estándar de Syslog disponible en la mayoría de dispositivos de conectividad y sistemas operativos, así como otras implementaciones más actuales y con características añadidas tales como Rsyslog[13], Syslog-NG[14] o Nxlog[15].

Los registros se pueden almacenar en ficheros texto o bases de datos. Para el análisis y visualización de la información es común utilizar un grupo de herramientas de manipulación de texto y búsqueda de patrones disponibles en la mayoría de las distribuciones de Linux. Los especialistas de seguridad y administradores de sistemas, frecuentemente ejecutan en la consola los comandos grep, sed, tail, head, cut, awk o perl entre otros, para procesar ficheros de trazas y obtener reportes(Cooper, 2014; Sobell, 2005; Albing, Vossen y Newham, 2007; Kemp, 2009).

Otro grupo de aplicaciones, son aquellas que incluyen herramientas para el almacenamiento e indexación y un sistema de visualización y generación de reportes,

[13] http://www.rsyslog.com/
[14] https://www.balabit.com/network-security/syslog-ng
[15] http://nxlog.org/

brindando una solución completa. En este grupo podemos destacar a ELSA[16], Graylog2[17] y Logstash ELK[18].

Existen otras aplicaciones como es el caso del visor de eventos de Windows[19], que permite la suscripción de máquinas clientes para hacer una copia local de los eventos generados.

Sistemas detectores de intrusiones de *host* como OSSEC trabajan bajo una arquitectura cliente servidor donde un agente lee y procesa los registros y los envía a un servidor central. Los sistemas SIEM mencionados en el epígrafe 1.2.1 pueden incluir soluciones completas para la gestión de trazas y eventos de seguridad.

Otra variante para la gestión centralizada es el caso de delegar el procesamiento y las herramientas en un tercero y contratar la infraestructura como servicio. Un ejemplo de este servicio es Loggly[20], que brinda diferentes acuerdos con los clientes para la gestión de trazas. En el epígrafe siguiente se hace una descripción más exhaustiva de cada una de las herramientas mencionadas.

1.8.1. Herramientas basadas en Syslog. Syslog-ng OSE, Rsyslog, Nxlog

Syslog-NG[21] ofrece una solución multiplataforma para la gestión de trazas mediante la utilización del protocolo Syslog. Dentro de sus principales características está la gran cantidad de opciones de filtrado que implementa y su facilidad de configuración. Posee una versión libre donde se han limitado varias funcionalidades y una versión empresarial de pago. En el Anexo 4, se listan algunas de sus funcionalidades.

Rsyslog[22] se ha convertido en el sistema por defecto de casi la totalidad de las distribuciones Linux. Los sistemas Unix/Linux manejan la gestión de eventos mediante la aplicación syslogd y sysklogd para los eventos generados específicamente por el núcleo del sistema operativo. Syslogd sólo permite el manejo de mensajes mediante el protocolo UDP, no implementa ningún tipo de mecanismo de fiabilidad en cuanto a la

[16] https://code.google.com/p/enterprise-log-search-and-archive/
[17] https://www.graylog.org/
[18] https://www.elastic.co/
[19] Soportado en Windows Vista o versiones superiores.
[20] https://www.loggly.com/
[21] http://www.balabit.com/network-security/syslog-ng
[22] http://www.rsyslog.com/

recepción de mensajes y tiene opciones de filtrado muy reducidas, teniendo en cuenta solamente la prioridad de los mensajes.

Rsyslog soporta múltiples módulos de entrada y salida. Permite filtrar y modificar los mensajes basado en reglas condicionales. La configuración mantiene la compatibilidad hacia atrás soportando los formatos descritos en la Tabla 1.9. La documentación disponible en línea describe en detalle cada una de las configuraciones[23].

Tabla 1.9. Tipos de configuración disponibles en Rsyslog.

Tipos de configuración	Descripción
Sysklogd	Formato estándar utilizado por Syslog
Legacy Rsyslog	Las declaraciones comienzan con el signo $. Formato disponible hasta la versión 6.
RainerScript	Nuevo formato disponible en versiones superiores y recomendado para configuraciones más complejas.

Según Rainer Gerhards, creador y desarrollador principal, Rsyslog permite manejar una gran cantidad de eventos. Según un artículo de 2010 donde plantea el paso de 40K mensajes por segundo hasta los 250K(Gerhards, 2010). El manejo de grandes volúmenes de mensajes y su disponibilidad en la mayoría de las distribuciones, son las características fundamentales que potencian su utilización en un sistema centralizado de recolección de trazas. Algunas de las características más significativas de Rsyslog pueden ser consultadas en el Anexo 5.

Nxlog está disponible en dos variantes, una propietaria denominada Edición Empresarial[24] y una de código fuente abierto denominada Edición Comunitaria[25]. Dentro de las características más notables está la correlación de eventos y tener en su variante comunitaria una versión compatible con los sistemas operativos Windows, con las mismas funcionalidades que en los sistemas GNU/Linux y UNIX.

Utiliza una arquitectura de módulos configurables como se muestra en la Tabla 1.7, que leen los eventos de múltiples entradas, procesan y dan formato a la información

[23] http://www.rsyslog.com/doc
[24] http://nxlog-ce.sourceforge.net/enterprise-edition
[25] http://nxlog-ce.sourceforge.net/

enviándola a múltiples módulos de salida. En el Anexo 6 se listan sus principales características.

Figura 1.7. Arquitectura base de Nxlog(Botond Botyanszki, 2009b).

Los tres sistemas cuentan con funcionalidades muy similares. Syslog-NG es el que presenta mayor cantidad de limitaciones en su versión libre. Rsyslog brinda la totalidad de sus funcionalidades y tiene la ventaja de estar soportado por la mayoría de las distribuciones de sistemas Linux, ampliamente utilizado en entornos de servidores. Nxlog tiene la ventaja a consideración del autor de la investigación de ser la herramienta libre más completa para la recolección de trazas disponible sistemas en operativos Windows.

1.8.2. OSSEC HIDS

OSSEC[26] es un detector de intrusiones de *host* (HIDS por sus siglas en inglés) escalable y multiplataforma bajo licencia GNU/GPL con las características siguientes:

- Posee un motor de correlación y análisis de trazas.
- Chequeo de integridad de ficheros.
- Monitoreo de registro de Windows.
- Detección de *Root Kits*.
- Alertas en tiempo real.
- Respuesta activa.

Ossec se utiliza como analizador de trazas para el análisis y el monitoreo de eventos generados por cortafuegos, IDSs, servidores web y registros de autenticación.

Se puede instalar en la mayoría de los sistemas operativos siguiendo perfiles de instalación disponibles:

[26] http://www.ossec.net/

- **Instalación local**: configuración para trabajar en un solo *host*.
- **Instalación como agente**: instalación del agente OSSEC en un *host* reportando hacia un servidor central.
- **Instalación como servidor**: recolecta y procesa los mensajes de cada uno de los agentes instalados.

Según el tipo de instalación OSSEC cumplirá distintas funcionalidades como se muestra en la Tabla 1.10(Bray, Cid y Hay, 2008a).

Tabla 1.10 Perfiles de instalación de OSSEC.

Instalación Local	Instalación como agente	Instalación como servidor
Chequeo de integridad	Chequeo de integridad	Chequeo de integridad local
Monitoreo de registro	Monitoreo de registro	Monitoreo de registro local
Respuesta activa	Respuesta activa	Respuesta activa local Centralización de alertas
Registro de trazas y generación de alertas		Centralización de trazas en formato Syslog

OSSEC permite analizar las trazas y detectar anomalías según un conjunto de preprocesadores y reglas realizando la función de detector de intrusiones de trazas (LIDS por sus siglas en ingles). La versión disponible, en el momento del desarrollo de la investigación, puede manejar 14 formatos de trazas incluyendo registros en más de una línea y salida de comandos.

Una de las principales características de OSSEC es la posibilidad de crear preprocesadores y reglas personalizadas para procesar nuevos formatos de trazas. De igual forma las reglas se pueden correlacionar y priorizar mediante estructuras condicionales.

Por otra parte, los eventos recolectados y generados por OSSEC son el resultado del procesamiento realizado por la aplicación. Si se requiere de almacenamiento de las trazas originales se debe combinar con otros mecanismos.

1.8.3. ELSA, Graylog2, Logstash ELK

Graylog2 es un sistema de gestión de trazas desarrollado en Java para el servidor y Ruby on Rails para la interfaz web. Permite recolectar mensajes vía Syslog ya sea TCP, UDP o utilizando un formato propio denominado Formato de trazas extendido Graylog(GELF por sus siglas en inglés). GELF no es más que datos JSON comprimidos en formato *zip*.

La interfaz web permite ver las trazas en una línea de tiempo, guardar las búsquedas realizadas, tener un registro de los *host* añadidos, integración de conectores, monitorear el estado del sistema, entre otras funcionalidades.

Graylog2 no provee ninguna forma alternativa para leer trazas desde ficheros según la documentación online[27] proponiendo la utilización de herramientas como Logstash.

El sistema empresarial de almacenamiento y búsqueda de trazas (ELSA por sus siglas en inglés) procesa mensajes utilizando Syslog-NG, Mysql para el almacenamiento y Sphinix[28] para la búsqueda e indexado a través de una interfaz web desarrollada en Perl que permite búsquedas asincrónicas por cadenas de texto arbitrarias, siguiendo el mismo concepto que los buscadores convencionales.

ELSA tiene dos mecanismos de instalación fundamentales: nodo sin interfaz web que funciona como agente de almacenamiento y retransmisión y nodo con interfaz web para realizar las búsquedas. Ambos componentes puede instalarse por separado comunicándose vía HTTPS o en un único servidor.

Logstash ELK es la unión de Logstash, Elasticsearch y Kibana para gestión centralizada de trazas. Logstash se compone de un marco integrado de recolección de trazas, centralización, almacenamiento y búsqueda. Tiene una gran variedad de posibilidades de formatos de entrada de trazas. Puede leer eventos vía TCP/UDP, desde ficheros, Syslog, eventos de Windows, entrada estándar (STDIN) entre otras fuentes.

Logstash se compone de tres módulos que incluyen decodificadores para distintas fuentes. Un registro de trazas pasa por cada uno de los módulos, realizándose las

[27] http://docs.graylog.org/en/1.0/pages/sending_data.html
[28] http://sphinxsearch.com/

transformaciones definidas como se muestra en la Figura 1.8. Los tres módulos principales van a tener las funcionalidades siguientes:

- Entradas: leen múltiples formatos de entrada, como ficheros, Syslog vía TCP o UDP, Unix, Zeromq, entre otros.
  - Las entradas se pueden procesar con codificadores que permiten estructurar la información en dependencia del formato de entrada como es el caso de netflow, JSON o ficheros multilínea.
- Filtros: permiten reorganizar, normalizar, añadir o eliminar información de los mensajes seleccionados utilizando patrones y expresiones regulares a través de la aplicación Grok(Turnbull, 2013).
- Salidas: permite almacenar o enviar los eventos procesados a múltiples destinos, como bases de datos, sistemas de representación gráfica, generación de notificaciones entre otros.

Figura 1.8. Esquema de los módulos configurables en Logstash.

En la versión 1.4.2 cuenta con 41 módulos de entradas con 20 decodificadores para determinados tipos de entradas, 50 tipos de filtros y la posibilidad de disponer de 55 formatos de salida, según la documentación disponible en el sitio web oficial[29]

Para ejecutar Logstash se requiere solamente contar con java instalado. Se distribuye como una aplicación única en formato jar, además se ofrecen las versiones empaquetadas en formato *deb* y *rpm*.

Elasticsearch es un motor distribuido de análisis y búsqueda desarrollado utilizando la librería Apache Lucene[30]. Se indexan y se pueden consultar contenidos siguiendo una

[29] http://logstash.net/docs/1.4.2/
[30] https://lucene.apache.org/core/

estructura tipo (clave, valor) accesibles vía HTTP. Todas las operaciones de almacenamiento y búsqueda utilizan el formato JSON.

Los datos se almacenan en una estructura denominada documento, que representa la unidad básica de información. Una colección de documentos se agrupa en índices. Los datos almacenados pueden ser de distintos tipos como enteros, cadenas, números de punto flotante, números IP, objetos, entre otros. Van a tener un identificador único, si no se define en el almacenamiento se asigna de forma dinámica al igual que los tipos de datos. Elasticsearch tiene una arquitectura de clúster por defecto. Una sola instalación se comporta como un clúster de un único nodo. Los índices pueden estar distribuidos en los nodos en estructuras denominadas bloques o *shards.*

Elasticsearch, al utilizar Lucene, soporta la búsqueda de términos utilizando operadores como *AND*, *ORD*, *NOT,* entre otros que forman parte de la sintaxis de la librería(Michael, Erik y Otis, 2010; Jayant Kumar, 2015). Otra de las características fundamentales es la utilización de un lenguaje denominado: Lenguaje Específico del Dominio (DSL por sus siglas en inglés). Las llamadas se hacen en el cuerpo de una petición HTTP siguiendo la estructura que se muestra en la Tabla 1.11. Las búsquedas en lenguaje DSL dan una gran flexibilidad y facilidad de lectura, por lo cual, es el método que debe ser utilizado para producción(Gormley y Tong, 2014; Kuć y Rogoziński, 2013).

Tabla 1.11. Estructura de las búsquedas DSL en Elasticsearch.

Estructura de una búsqueda DSL	**Ejemplo de búsqueda**

GET /_search
{
QUERY_NAME: {
ARGUMENT: VALUE,
ARGUMENT: VALUE,...
}
}
	GET /_search
	{
	"query": {
	"match_all": { }
	}
	}

Elasticsearch cuenta con un gran número de características y funcionalidades orientadas al almacenamiento de grandes volúmenes de datos, que no respondan a un esquema rígido y donde se requiera de realizar búsquedas cercanas a tiempo real.

Kibana se encarga de la visualización y de proveer la interfaz de búsqueda para los registros de trazas almacenados en Elasticsearch. Tiene una interfaz personalizable donde se pueden añadir diferentes áreas de trabajo. Permite la creación de tablas, gráficos y múltiples tipos de visualización para los resultados de las búsquedas realizadas(Turnbull, 2013). En su versión 4.1 posibilita representar la información almacenada mediante ocho tipos de gráficos configurables. El resultado de una búsqueda puede ser configurado para su representación en el tiempo utilizando los gráficos que se describen en la Tabla 1.12.

Tabla 1.12. Paneles disponibles en Kibana para la visualización de la información.

Panel	Descripción
Gráfico de área	Representa los datos sombreando el área bajo la curva. Con contornos rectos o suavizados.
Tabla de datos	Muestra los datos en formato tabular en una tabla.
Gráfico de líneas	Representa los datos uniendo los puntos con una línea. Se configuran contornos rectos o suavizados.
Panel de texto	Panel que acepta entradas de texto en el formato *GitHub-flavored Markdown*[31].
Métrica	Muestra un número que representa la agregación seleccionada
Gráfico de pastel	Representa en un gráfico de pastel las agregaciones seleccionadas
Mapa	Representa los datos de la agregación utilizando círculos en un mapa según los datos de coordenadas geográficas obtenidas
Gráfico de barras verticales	Agregación de datos en un gráfico de barras vertical

La configuración de los gráficos descritos conforman los distintos paneles de mando para el análisis de las trazas almacenadas por parte de los especialistas. Cada panel es configurable y permite la lectura de datos de distintos tipos de índices(Elasticsearch, 2015).

Artyom Churilin(Artyom Churilin, 2013), realiza un análisis comparativo de las herramientas descritas donde expone las ventajas y desventajas de cada solución. En

[31] https://help.github.com/articles/github-flavored-markdown/

su análisis, con el que concuerda el autor de esta investigación, propone Logstash ELK integrado con Rsyslog como la solución más factible en cuanto al compromiso de usabilidad y rendimiento. Expone según la utilización de Graylog2 como una herramienta más factible para entornos donde predomine un único tipo de traza.

En el caso de ELSA, aunque destaca el rendimiento y las funcionalidades de generación de alertas y tickets para la respuesta a incidentes, tiene las limitantes de estar basado en Syslog-NG en cuanto las diferencias entre la versión empresarial y comunitaria, y las posibilidades que brinda en cuanto al monitoreo, funciones de análisis y visualización.

1.8.4. Sistemas de gestión de trazas como servicio

La gestión de trazas como servicio consiste en contratar toda la infraestructura a un proveedor de servicios en la nube que se encargue del almacenamiento, procesamiento y visualización. Tiene como principal ventaja el ahorro en cuanto equipamiento, montaje, instalación y mantenimiento de software y personal especializado.

Tabla 1.13. Planes libre de costo ofrecidos por sistemas de gestión de trazas en la nube.

Herramientas	Cantidad de eventos	Retención de datos	Cantidad de fuentes de datos
Logsense	1 Millón eventos	Configurable acorde al plan[32]	Ilimitado
Papertrail	100MB/Mes	7 Días	No especifica
Logentries	5GB/Mes	7 Días	No especifica
Sumologic	500MB/Día	7 Días	No especifica
Loggly	Limitado por día[33]	7 Días	Múltiples fuentes, tres grupos

La gestión de trazas como servicio requiere de una conexión a Internet estable y con un ancho de banda que permita manejar el flujo de trazas que se quiere enviar. Para flujos considerables de datos es necesario establecer acuerdos de pago para el

[32] El periodo de retención de datos debe ser tal que no exceda la cantidad de eventos.
[33] Una vez que se llegue al valor límite se detiene el indexado y continua al día siguiente, hora UTC.

servicio y se debe tener en cuenta las implicaciones de seguridad al enviar datos hacia un almacenamiento en la nube.

1.9. Conclusiones parciales

Del análisis realizado en el capítulo, se revisó la guía SP800-92 del NIST, destacándose la necesidad, respecto a las recomendaciones analizadas, de revisar nuevas herramientas y tipos de trazas. Se valoraron los componentes fundamentales y los eventos de mayor importancia a almacenar y analizar en un sistema de gestión de trazas de seguridad acorde a los estándares y regulaciones más utilizados. Se analizaron los distintos formatos de trazas y herramientas de procesamiento y visualización, donde se evidenció el uso de Syslog como protocolo, en una arquitectura centralizada, manejando las trazas con un formato estructurado. De las soluciones de gestión de trazas analizadas, según sus características, Logstash ELK resulta la más factible para una implementación centralizada. Como elemento esencial, se evidenció la necesidad de una solución que integre eventos, protocolos y herramientas en un único sistema.

Capítulo 2. Marco de trabajo para la gestión centralizada de trazas de seguridad utilizando herramientas de código abierto

En el capítulo siguiente se propone un marco de trabajo para la gestión centralizada de trazas de seguridad. Se lleva a cabo la selección de las principales herramientas para obtener las fuentes de trazas que deben ser recolectadas acorde a las principales regulaciones, normas y guías de buenas prácticas. Se propone la arquitectura de despliegue correspondiente y cada uno de los componentes de software que la integran.

2.1. Marco de trabajo propuesto para la gestión centralizada de trazas de seguridad

Partiendo de la no existencia de una solución integradora, se propone en la presente investigación, un marco de trabajo para la gestión centralizada de trazas de seguridad. El marco de trabajo, representado en la Figura 2.1, lo integran los procesos de planeación, diseño y ejecución necesarios para poner en operación una arquitectura centralizada de gestión de trazas. Como primer componente se requiere definir políticas, roles y responsabilidades que sustenten la ejecución centralizada del procesamiento de las trazas. La arquitectura propuesta debe estar correctamente dimensionada mediante el cálculo de los parámetros para la estimación del volumen de trazas y almacenamiento.

La arquitectura se compone de cuatro capas:

- Generación de trazas.
- Transporte.
- Análisis y almacenamiento.
- Monitoreo.

Dentro de las funciones generales están:

- Acciones de configuración necesaria para el filtrado de los eventos generados.

- Acciones de normalización, conversión y reducción de los eventos recibidos y la agregación en una localización central mediante la compresión e indexado de las trazas.

Una vez almacenados los registros, una aplicación de monitoreo se encarga de brindar las funciones de visualización y generación de variables estadísticas asociadas a las trazas que se presentan al analista de seguridad.

Cada una de las capas descritas se integran con un conjunto de configuraciones de seguridad de forma total o parcial que incluyen:

- Utilizar un servidor de tiempo para mantener sincronizada la hora de las diferentes fuentes de generación, almacenamiento y sistemas de monitoreo de trazas.
- Utilización de buffers de almacenamiento temporal para evitar la pérdida de eventos ante un corte de transmisión, recepción o problemas con el canal de red.
- Utilización de canales cifrados para la transmisión de datos (TLS) y protocolos seguros para la transmisión (RELP, TCP) siempre que sea posible.
- Generación de funciones resumen y chequeo de integridad de los ficheros de trazas almacenados del lado del servidor.
- Monitoreo del acceso a los ficheros de trazas registrando las operaciones, lectura, escritura, creación o modificación que puedan realizarse a nivel de sistema de ficheros.

En el marco de trabajo propuesto, cada una de las capas de la arquitectura se compone de un grupo de herramientas que trabajan de forma integrada. La utilización de varias herramientas y su instalación requiere del manejo de múltiples ficheros de configuración. Para esto se propone que la gestión de configuración se realice de forma automatizada.

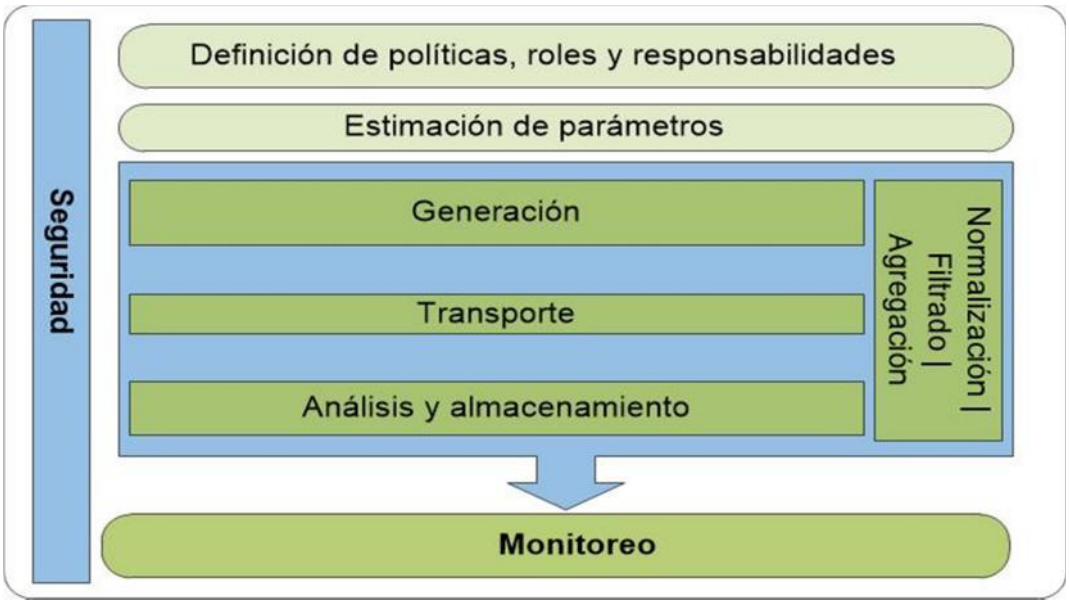


Figura 2.1. Marco de trabajo para la gestión centralizada de trazas de seguridad.

2.1.1. Definición de políticas, roles y responsabilidades

Acorde a la guía del NIST SP800-92 descrita en el Epígrafe 1.2.1, cada una de las capas de la arquitectura propuesta debe estar respaldada por políticas que enmarquen la función que van a desempeñar. A continuación, se listan los aspectos a tener en cuenta por cada una de las capas y que deben servir como base para la elaboración de políticas.

- Generación de trazas:
  - Tipos de *host* que debe realizar el registro de trazas.
  - Componentes de los sistemas que deben realizar el registro de trazas.
  - Campos que deben registrarse por cada tipo de eventos.
  - Frecuencia con que se registran los eventos.
- Transporte:
  - Tipos de *hosts* que deben transferir las trazas hacia una localización central.
  - Qué componentes de los sistemas deben realizar el registro a una localización central.
  - Cómo debe realizarse la transferencia.
  - Con qué frecuencia se transmiten las trazas.
  - Cómo se garantiza la seguridad de las trazas en el transporte.

- Almacenamiento y borrado:
  - Cómo se garantiza la seguridad de las trazas almacenadas.
  - Por cuánto tiempo se mantienen las trazas almacenadas.
  - Cómo se eliminan los registros no necesarios.
  - Con que regularidad se borran las trazas.
- Análisis y visualización:
  - Con qué frecuencia se analizan las trazas.
  - Quién tiene acceso a las trazas almacenadas y cómo se registran los accesos.
  - Qué acciones deben ser tomadas en cuenta cuando se detecta una anomalía o evento de seguridad.
  - Cómo se protege la confidencialidad, integridad y disponibilidad tanto de los análisis realizados como de los resultados obtenidos.

Además de las pautas planteadas en la guía, se debe incluir en la generación, las fuentes que permitan obtener los registros más significativos acorde al análisis realizado en el Epígrafe 1.4.

La definición de roles que se propone parte de la base de dos grupos primarios integrados por los especialistas de infraestructura y sistemas y los especialistas de seguridad informática:

Tabla 2.1. Definición de roles y responsabilidades.

Rol	Descripción
Administrador de infraestructura y sistemas	Responsable de la instalación configuración y mantenimiento de los sistemas y servicios incluyendo el sistema de gestión de trazas
Especialistas de seguridad informática	Responsable de la auditoría, análisis de información de seguridad, monitoreo de la información recolectada, generación de reportes, búsquedas, personalización y extensión del sistema de gestión de trazas.

En dependencia del tamaño y complejidad de la organización, se tendrá en cuenta la cantidad de personal necesario en cada rol, partiendo de la base de dos especialistas como mínimo.

2.1.2. Estimación de parámetros. Cálculo de espacio de almacenamiento

Los parámetros principales a tener en cuenta son:

- Cálculo del espacio necesario para almacenar las trazas recolectadas.
- Análisis del ancho de banda y el impacto sobre el tráfico de la transmisión en línea de las trazas.

En dependencia de la localización geográfica de las fuentes de trazas se debe tener en cuenta el consumo de ancho de banda en el proceso de trasmisión. Si es posible la red de gestión debe estar separada de la red de datos, en otro caso, su influencia debe ser mínima. Si el tráfico asociado al almacenamiento de trazas influyera sobre el flujo de datos de los servicios, se debe considerar: disminuir la cantidad de fuentes que se recolecten en línea, priorizar qué eventos deben enviarse o planificar la copia fuera de línea en horarios de menor carga.

El espacio de almacenamiento dedicado va a depender de la cantidad de fuentes de trazas que se hayan configurado y del tamaño en bytes de las trazas generadas.

La aproximación más utilizada para medir la cantidad de trazas generadas consiste en medir la cantidad de eventos por segundos (EPS por sus siglas en inglés)(Butler, 2009).

$$EPS = \frac{No.\,de\ eventos\ del\ sistema}{Período\ de\ tiempo\ en\ segundos}$$

Para calcular la cantidad de gigabytes de datos generados diariamente:

$$GB_{día} = \frac{EPS\ x\ Tamaño\ de\ una\ línea\ en\ bytes}{1073741824}\ x\ 64800$$

La cantidad de gigabytes diarios multiplicado por el periodo de tiempo que se retengan las trazas, determinará el espacio total de almacenamiento necesario.

El tamaño de una línea de trazas varía mucho en dependencia del tipo de registro que se almacene. Para determinar el tamaño de una línea en bytes se puede calcular la cantidad de caracteres promedio por línea de trazas y multiplicarlo por la cantidad de bytes que ocupa un carácter. La mayoría de las trazas se escriben en inglés. Según la RFC 3629(F. Yergeau, 2003) un carácter UTF-8 puede tomar de uno a cuatro bytes. Utilizar uno o dos bytes como valor por defecto será suficiente según la opinión del autor de la investigación.

La cantidad de caracteres por cada línea de traza se obtiene del conteo promedio de cada línea de los registros almacenados en un período de tiempo. En el Anexo 7 se incluye el código de una aplicación para obtener los datos necesarios para calcular el tamaño promedio de una línea de trazas.

El espacio total puede reducirse compactando las trazas. El formato *gzip* es uno de los más utilizados. *Gzip* Permite la compresión y lectura en línea lo que posibilita no tener que descomprimir los ficheros si es necesario hacer algún tipo de operación de lectura. Skibiński y Swacha en el artículo: Compresión rápida y eficiente de trazas(Skibiński y Swacha, 2007), proponen trasformaciones a las trazas para aumentar la eficiencia de compresión. Mencionan una tasa de compresión de los ficheros de muestra originales para distintos ficheros de trazas entre 1:8 y 1:29. En el Anexo 8 se analizan las trazas de un mes para cuatro aplicaciones: Qmail, Openfire, Squid y Nginx obteniéndose una tasa de compresión promedio entre 1: 10 y 1:14. El autor de la investigación considera suficiente tomar una tasa de compresión aproximado de 1:10 a la hora de determinar el espacio total necesario.

2.2. Arquitectura de despliegue

El sistema se estructura mediante la arquitectura que se muestra en la Figura 2.2. Está compuesto por las herramientas que permiten recolectar los distintos tipos de eventos junto con el resto de los componentes necesarios para implementar el marco de trabajo propuesto. Se propone una solución de gestión de trazas apoyada en herramientas de software de código abierto que permiten gestionar todo el proceso de recolección, transporte, almacenamiento, generación de reportes sincronización de tiempo y gestión de configuración.

La Tabla 2.2 describe de forma general cada una de las herramientas que se utilizan en la arquitectura propuesta.

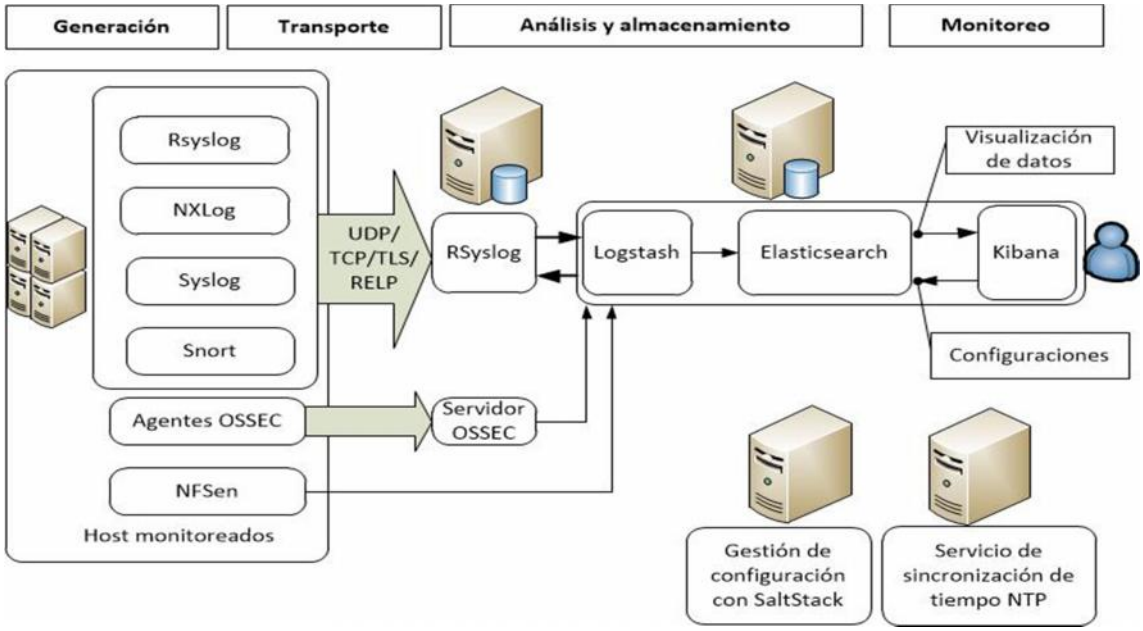


Figura 2.2. Arquitectura de despliegue para la gestión centralizada de trazas de seguridad.

Dentro de las herramientas descritas es importante señalar la presencia de Auditd. Esta herramienta no es necesario instalarla, viene preinstalada con el sistema operativo. Para el caso de Windows, se incluye la misma funcionalidad dentro de las políticas del sistema.

Tabla 2.2. Selección de herramientas que componen la arquitectura propuesta

Software	Descripción
Rsyslog	Software para el procesamiento de trazas en los sistemas Linux. En la mayoría de las distribuciones aparece como opción predeterminada. Recolecta las trazas del sistema permitiendo la recolección, almacenamiento, filtrado, reducción y retransmisión de los eventos procesados.
Nxlog	Software para el procesamiento de trazas. Recolecta las trazas de diversas fuentes. Implementa múltiples mecanismos de filtrado, permite correlacionar eventos, permite el almacenamiento y retransmisión en múltiples formatos y es multiplataforma estando disponible para Linux, UNIX, Windows y Android(Botond Botyanszki, 2009a).

Snort	Sistema de Detección de Intrusiones de Red (NIDS por sus siglas en inglés) con capacidad de análisis y registro de paquetes en tiempo real(Baker, Esler y Alder, 2007).
Ossec	Sistema de Detección de Intrusiones de *Host* (HIDS) con capacidades de análisis de trazas, chequeo de integridad de ficheros, monitoreo de políticas, detección de rootkits, generación de alertas en tiempo real y respuesta activa(Bray, Cid y Hay, 2008b).
Auditd	Mecanismo en los sistemas Unix/Linux para registrar información relevante a la seguridad basado en reglas(Miclea, 2012).
Logstash	Herramienta para el manejo de trazas. Permite la recolección en múltiples formatos, gran cantidad de funciones de filtrado, y múltiples formatos de salida(Gormley y Tong, 2014).
Elasticsearch	Motor distribuido de análisis y búsqueda. Se integra con la mayoría de los lenguajes más populares y se maneja casi completamente mediante utilizando JSON sobre HTTP. Actualmente se integran Logstash + Elasticsearch + Kibana para proveer una solución completa de análisis de trazas conocida como ELK(Kuć y Rogoziński, 2013; Gormley y Tong, 2014).
Kibana	Sistema de visualización que permite realizar y visualizar múltiples tipos de búsquedas en datos almacenados sobre Elasticsearch(Gormley y Tong, 2014).
NTP	Protocolo designado para sincronizar la hora de las computadoras y dispositivos conectados a la red(Rybaczyk, 2005).
Nfsen	Nfsen lo componen un conjunto de herramientas en línea de comandos que permiten capturar tráfico netflow. Lo componen nfcapd, nfdump, nfprofile, nfreplay, nfclean.pl, ft2nfdump(Fry y Nystrom, 2009)
SaltStack	SaltStack permite la gestión centralizada y la configuración, instalación de software y ejecución de comandos en un gran número de computadores, servidores, estaciones de trabajo y dispositivos de forma simultánea. Una misma configuración puede ser aplicada a múltiples arquitecturas y sistemas operativos mediante la interpretación de plantillas genéricas (Craig Sebenik, 2015)

La arquitectura mostrada envía los eventos hacia un servidor central de Syslog, después de ciertos niveles de procesamiento y reducción envía las trazas hacia Logstash para un segundo nivel de filtrado, normalización, indexado y almacenamiento

en el sistema de indexado Elasticsearch. Una vez almacenadas se puede realizar y mostrar los resultados de distintos tipos de búsquedas utilizando Kibana. Las trazas almacenadas pueden tener indicadores como nombres de índices diferentes. Kibana puede ser configurado con múltiples paneles de mando (Dashboard) asociados a los datos almacenados o a parte de estos. En el caso del procesamiento de tráfico Netflow se envía directamente hacia Logstash y para Ossec se utiliza un servidor central donde se recolectan las alertas recibidas antes de enviarlas.

Al igual que Rsyslog, Logstash permite la recepción de eventos directamente. Aquellos eventos que sean dirigidos a Logstash y se haya definido su retención a largo plazo pueden redirigirse hacia Syslog.

Por otro lado, Rsyslog implementa la posibilidad de enviar los eventos directamente hacia Elasticsearch, aunque no posee las posibilidades de filtrado y procesamiento de Logstash.

El servidor Syslog va a almacenar las trazas a largo plazo, cumpliendo con la política de retención que se haya definido mientras que el almacenamiento en Elasticsearch responde al almacenamiento a corto plazo de las trazas indexadas para permitir el monitoreo y la generación de información en periodos cercanos a tiempo real.

2.2.1. Generación

Las herramientas descritas van a permitir, una vez configuradas, recolectar y procesar los registros de eventos obtenidos en la Tabla 1.4. En la Tabla 2.3 se relaciona las principales herramientas con cada uno de los registros obtenidos.

Tabla 2.3. Funcionalidad de cada una de las herramientas de recolección de trazas y su relación con los principales registros de eventos que deben recolectarse.

Registro de evento	Herramientas	Funcionalidad asociada
Acciones con privilegios administrativos	Lectura de eventos de Windows con Nxlog, Syslog, Rsyslog	Recolección de eventos de auditoría del sistema
Acceso a las trazas	OSSEC, Auditd	Chequeo de integridad y de

		acceso de llamadas al sistema
Monitoreo de sesiones	Lectura de eventos de Windows con Nxlog, Rsyslog	Se registran en los eventos de auditoría del sistema
Eventos satisfactorios y fallidos de usuarios, aplicaciones y sistemas	Lectura de eventos de Windows con Nxlog	Recolección de eventos de auditoría del sistema
Creación, modificación y borrado de objetos del sistema	Lectura de eventos de Windows con Nxlog, Auditd	Acceso de llamadas al sistema
Monitoreo de fuga de información	Snort	Reglas de análisis de tráfico que detecten cadenas y patrones determinados
Registro de conexiones remotas	Syslog, Nxlog	Recolección de eventos de auditoría de las aplicaciones
Registro de trazas tráfico de red	Netflow, Nfsen	Transporte y almacenamiento del tráfico capturado
Eventos Críticos	Nxlog, Rsyslog, Syslog	Recolección de eventos de auditoría del sistema

Las herramientas seleccionadas requieren de configuraciones básicas que permitan su integración en la arquitectura propuesta y la posibilidad de recolectar y procesar los registros de eventos analizados.

En Rsyslog se configuran módulos de entrada que recolectan y encolan los mensajes. Posteriormente se definen acciones que se ejecutan para cada mensaje donde se pueden modificar los campos. Una de las funcionalidades principales es la opción de crear plantillas para almacenar los ficheros de forma organizada.

Mediante este mecanismo también se pueden almacenar los registros en un gestor de bases de datos. Rsyslog puede funcionar como agente de retransmisión enviando los eventos mediante los protocolos UDP, TCP o RELP. El protocolo RELP se encarga de la transmisión fiable de eventos añadiendo chequeos en la entrega de mensajes a nivel de aplicación(Rainer Gerhards, 2014b).

La creación de plantillas incluye variables dinámicas que forman parte del cuerpo del mensaje Syslog en combinación con cadenas de texto.

Tabla 2.4. Configuración de una plantilla dinámica en Rsyslog y su utilización en una estructura condicional.

Plantilla de configuración parcial en Rsyslog para el almacenamiento de registros

```
$template DYNsecure,"/var/log/%HOSTNAME%/secure"
if \
      $source != 'localhost' \
            and \
      $syslogfacility-text == 'authpriv' \
then    ?DYNsecure
```

En el ejemplo de configuración mostrado en la Tabla 2.4, se define una plantilla donde los mensajes de seguridad van a ser almacenados en el directorio */var/log/* y a continuación se crea un directorio con el nombre del *host* que envía los eventos. Para garantizar que sólo los eventos de seguridad se almacenen siguiendo la plantilla creada, se programa un filtro condicional que aplica a los eventos de tipo authpriv.

Con Nxlog se leen los registros de trazas generados por Windows y se envían en formato Syslog encapsulando el contenido en el campo mensaje como una estructura JSON. Los eventos que las aplicaciones generan de forma independiente hacia un fichero texto se procesan con el módulo *im_file* y los eventos de Windows con *ms_vistalog*. El módulo *ms_vistalog*, permite en la configuración, añadir filtros de búsqueda utilizando una estructura en formato XML equivalente a la descrita por la documentación de Microsoft [34]. En la Figura 2.3 se muestran los bloques de configuración en Nxlog y en el Anexo 9 se muestra el fichero de configuración completo.

[34] Event Queries and Event XML. https://msdn.microsoft.com/en-us/library/bb399427%28v=VS.90%29.aspx

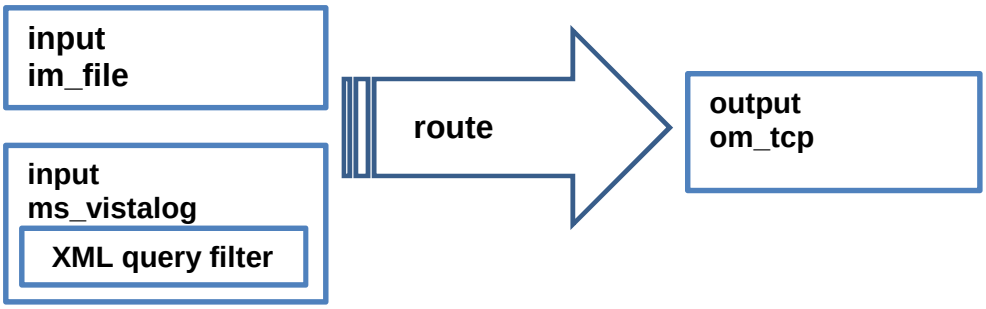


Figura 2.3. Bloques de configuración de Nxlog para la lectura y envío de eventos en Windows.

Snort fue desarrollado por Martin Roesch(Baker, Esler y Alder, 2007). Es capaz de llevar a cabo el análisis de tráfico y registro de paquetes en tiempo real sobre el protocolo de Internet IP. Puede realizar análisis de protocolo, búsqueda y emparejamiento de contenidos así como la detección de varios tipos de ataques. Trabaja fundamentalmente mediante la utilización de firmas, o como se conoce generalmente, basado en reglas(Caswell, Beale y Baker, 2007).

Las reglas son provistas en línea desde el propio sitio web para usuarios subscritos o disponibles para descargar después de un plazo de 30 días. Alternativamente también ofrecen ficheros de reglas de descarga gratuita en otros sitios para la comunidad, como es Emerging Threats[35].

Snort se puede instalar desde los fuentes o utilizando los ficheros binarios disponibles. La mayoría de las distribuciones de UNIX/Linux lo incluyen dentro de sus repositorios posibilitando la instalación siguiendo el mecanismo de gestión de paquetes asociado a la distribución.

Cuando se instala, los ficheros de configuración se ubican en el directorio */etc/snort/.* En este directorio se ubica *snort.conf* donde se define la localización de las reglas, comúnmente en el directorio */etc/snort/rules*/. Las reglas se nombran según la fuente y la función, ya sean reglas relacionadas con tráfico de servicios web, SQL, FTP o relacionadas con malware, escaneos, entre otras. Dentro del directorio *rules* se ubica un fichero, sin reglas asociadas, denominado *local.rules* donde se pueden añadir

[35] ETOpen Ruleset. http://www.emergingthreats.net/open-source/etopen-ruleset

reglas personalizadas según las políticas definidas en la institución y características particulares del entorno de red.

Un ejemplo de regla se muestra a continuación donde se detecta y se alerta sobre tráfico asociado a Google Talk.

alert tcp $HOME_NET any -> $EXTERNAL_NET 443 (msg:"ET POLICY Google Talk TLS Client Traffic"; flow:established,to_server; content:"gmail.com"; nocase; content:"jabber"; nocase; distance:64; within:78; reference:url,talk.google.com; reference:url,www.xmpp.org; reference:url,doc.emergingthreats.net/2002330; classtype:policy-violation; sid:2002330; rev:4;)

Reglas similares o con patrones de búsqueda equivalentes pueden ser utilizadas para detectar tipos de tráfico que violen las políticas establecidas en cuanto al manejo de información. Snort ante la detección de un evento puede ejecutar acciones de prevención bloqueando el tráfico asociado.

Ossec se configura para el chequeo de integridad, donde se verifica diariamente el estado de todos los ficheros que se encuentran en /etc, /usr/bin, /usr/sbin, /bin, /sbin. Cualquier cambio genera una alerta(Bray, Cid y Hay, 2008b). A las configuraciones de Ossec se pueden agregar manualmente cualquier fichero que se desee monitorear.

Auditd es la herramienta en el espacio de usuario para acceder a las funciones de auditoria del kernel de Linux.

Audit permite monitorear las acciones siguientes:

- Monitorear el acceso a los ficheros.
- Monitorear las llamadas al sistema.
- Registrar comandos ejecutados por el usuario.
- Registrar eventos de seguridad.
- Monitorear eventos de acceso a la red.

Audit puede registrar 141 tipos de eventos en las trazas generadas. La guía de seguridad de RedHat describe cada uno de los tipos generados(Robert Krátký, Yoana Ruseva y Tomáš Čapek, 2015).

Las acciones de monitoreo se especifican mediante reglas definidas en el fichero */etc/auditd/audit.rules*. Una regla para monitorear cambios de contraseña en el sistema sería:

auditctl -w /etc/passwd -p wa -k passwd_changes

En este caso se van a registrar todos los accesos de escritura y modificación de atributos del fichero *password* y la traza registrada contendrá la cadena *passwd_changes*.

2.2.2. Almacenamiento a largo plazo en el servidor Syslog e indexado en Elasticsearch

La arquitectura propuesta en la Figura 2.2 consta de dos niveles de almacenamiento. El servidor central donde se almacenan todas las trazas será el principal y donde es necesario reservar la mayor cantidad de espacio. El segundo nivel de almacenamiento, es a corto plazo, para el monitoreo y generación de reportes. Los ficheros de trazas se almacenan utilizando plantillas dinámicas que permiten la rotación utilizando la fecha y la hora actual como parte del nombre de los directorios y ficheros.

En la Figura 2.4 se muestra la estructura propuesta que va a seguir el almacenamiento central. Las entradas asociadas a año, mes y día son variables dinámicas que se van creando según la fecha actual y garantizan la rotación de los ficheros para cumplir con la política de retención que se haya establecido. Cada directorio representa una categoría genérica que va a tener varías subcategorías y el fichero de trazas en un tercer nivel. Ejemplo de subcategorías pueden ser: para el caso de servidores, dividido por tipo de sistema operativo y para el caso de hardware, por el tipo de dispositivo.

La cantidad de categorías y subcategorías va a depender de las particularidades del entorno de red donde se aplique la arquitectura. Una vez almacenadas las trazas el proceso de compresión permite optimizar el espacio disponible. El sistema propuesto, para la compresión recorre recursivamente los directorios y selecciona los ficheros anteriores al día actual. Se programa como una tarea diaria. En el Anexo 10 se incluye el código de una aplicación para la ejecución de la tarea.

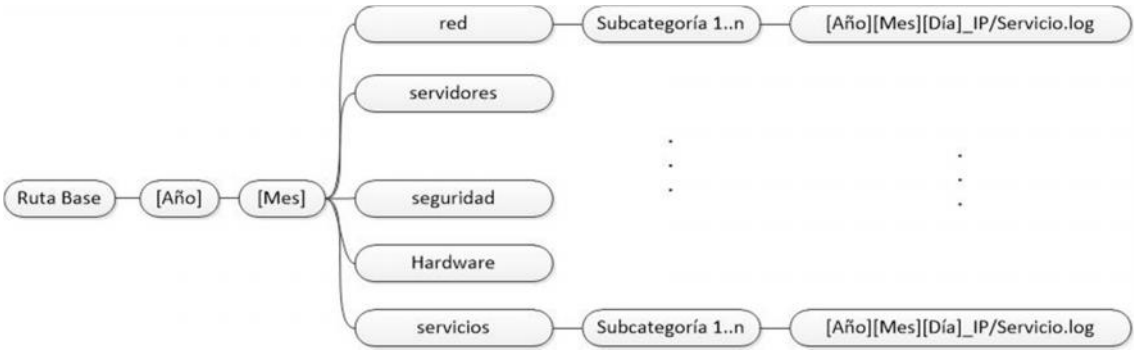


Figura 2.4. Estructura de directorios para el almacenamiento de las trazas en el servidor central de Syslog.

El almacenamiento a corto plazo se realiza en el servidor Elasticsearch. Las trazas se reciben en Logstash y van a estar almacenadas he indexadas para facilitar las búsqueda y generación de reportes, ya sea directamente mediante la interfaz de Kibana o accediendo a través de las funciones que provee el servidor. En el Anexo 11 se puede ver un ejemplo de configuración de Logstash para la recepción y almacenamiento, específicamente de los registros de Windows.

El espacio de almacenamiento va a estar determinado por los factores siguientes:

- Cantidad de campos que sean indexados y almacenados.
- Espacio de tiempo que se desee mantener los registros.

Elasticsearch recrea su estructura propia en el sistema de ficheros para el almacenamiento de los registros. La estructura de nombres de índices propuesta sería según el formato:

Categoría_subcategoría_[año][mes][día]

Siguiendo esta nomenclatura se mantiene una correspondencia con las trazas almacenadas a largo plazo y se crea un índice diario lo que facilita el proceso de borrado una vez que se ocupe el espacio de almacenamiento asignado.

Para optimizar el almacenamiento utilizado, se debe tener en cuenta como se crean los índices. Jordan Sissel[36], expone un conjunto de pruebas donde el almacenamiento,

[36] Jordan Sissel. Desarrollador principal de la aplicación de gestión de trazas Logstash y actualmente parte de Elastic. Compañía que brinda la solución completa de Elasticsearch, Logstash y Kibana

siguiendo la configuración por defecto puede tomar de cuatro a seis veces el tamaño del fichero de trazas en texto plano y propone un conjunto de cambios que reducen la utilización de espacio de 1.5 a 1.6 veces. La base de los cambios está en eliminar entradas duplicadas en el indexado. Particularmente los campos: @_all, @message, @source(Jordan Sissel, 2012).

2.2.3. Monitoreo

El monitoreo se realiza a través de la interfaz de Kibana. Se leen los índices existentes en Elasticsearch y automáticamente en la interfaz se listan todos los campos de datos disponibles de los documentos almacenados.

La información de configuración asociada a los paneles de mando creados se almacena en Elasticsearch en un índice especial con el nombre *.kibana.*

En la Interfaz web se siguen los pasos que se describen para la creación de los paneles de visualización.

- En el menú de configuraciones se selecciona la lista de índices que se hayan añadido desde Logstash en Elastisearch.
- En el menú de visualización se añaden los gráficos con los datos y agrupaciones que se quieran mostrar.
- En el menú de paneles de mando se crea un panel nuevo y se añaden los gráficos creados. Desde este menú se pueden crear múltiples paneles con los gráficos interactivos que se hayan configurado en el menú de visualización.

Desde la interfaz principal de Kibana se pueden seleccionar varios rangos de tiempo para la visualización. En el Anexo 12 se muestra la interfaz de Kibana con la ubicación de los menús principales.

2.2.4. Sincronización de tiempo

Cada dispositivo en la red debe tener configurado la fecha y hora correcta para que no haya inconsistencias en la información de las trazas que se generan y almacenan centralmente.

El protocolo NTP, permite la sincronización entre los dispositivos. Para esto es necesario que exista un servidor central de tiempo y que los clientes lo encuesten regularmente.

La configuración de los clientes puede llevarse a cabo mediante tareas programadas o gestionando la configuración como se propone en el epígrafe 2.2.5.

2.2.5. Gestión de configuración

El proceso de gestión de configuración permite automatizar y acelerar la instalación y configuración de software y sistemas mediante la orquestación de tareas comunes.

Para la gestión centralizada de trazas es necesario modificar la configuración de los sistemas en la red para que envíen la información hacia un servidor central. La utilización de un sistema de gestión de configuración permite mantener una configuración única, para todos los clientes, manejada centralmente.

Las configuraciones más comunes que se manejan son:

- Sincronización de tiempo.
- Configuración de los agentes Syslog para el envío de trazas hacia el servidor central.
- Monitoreo del estado de los servicios asociados a generación y recolección de trazas garantizando que se encuentren activos.

Para la gestión de configuración se propone SaltStack. La herramienta permite la gestión centralizada y la configuración, instalación de software y ejecución de comandos en un gran número de computadores, servidores, estaciones de trabajo y otros dispositivos de forma simultánea. Una misma configuración puede ser aplicada a múltiples arquitecturas y sistemas operativos mediante la interpretación de plantillas genéricas (Craig Sebenik, 2015).

2.3. Seguridad de las trazas en la arquitectura centralizada

Garantizar la seguridad de las trazas en el proceso de generación, transmisión y almacenamiento es una tarea fundamental. Una de las primeras acciones de un

atacante es el borrado de sus huellas en el sistema. Mantener la disponibilidad, integridad y confidencialidad sobre los registros requiere de protocolos y configuraciones específicas.

Se debe garantizar la disponibilidad evitando la pérdida de mensajes siempre que la aplicación que envía las trazas lo permita. Tanto Rsyslog como Nxlog tienen mecanismos para almacenar temporalmente las trazas hacia disco en caso de que se pierda la conexión con el servidor al que se envían. En la Tabla 2.5 se muestra la configuración necesaria para ambas aplicaciones.

Tabla 2.5. Configuraciones de Rsyslog y Nxlog para prevenir la pérdida de mensajes.

Rsyslog	Nxlog
$WorkDirectory /var/spppl/	<Processor buffer>
$ActionQueueFileName	Module pm_buffer
$ActionQueueMaxDiskSpace	# 1Mb buffer
$ActionQueueSaveOnShutdown	MaxSize 1024
$ActionQueueType	Type Mem
$ActionResumeRetryCount -1	# warn at 512k
. @@remote-host:514	WarnLimit 512
	</Processor>

La transmisión de eventos debe realizarse, siempre que sea posible, utilizando el protocolo TCP. Este ofrece mayor garantía de que no ocurran pérdidas de paquetes. Si la transmisión viaja por canales con alto grado de riesgo, tanto en Nxlog como en Rsyslog se pueden configurar conexiones cifradas mediante TLS según las especificaciones en la RFC5425(Cisco Systems, Inc., 2009).

La confidencialidad de las trazas es un tema bastante complejo de abordar, principalmente por la diversidad de formatos de trazas existentes, por los grandes volúmenes que son necesarios almacenar y por la necesidad de poder realizar búsquedas y reportes en el menor tiempo posible. Entre las soluciones clásicas están: la distribución de las trazas en varias localizaciones, imprimirlas en hojas de papel continuo o utilizar dispositivos de Múltiple lectura y escritura sólo una vez (WORM por sus siglas en inglés). Otras soluciones propuestas pueden ser consultadas en (Ma y Tsudik, 2009; Schneier y Kelsey, 1998; Bellare y Yee, 1997). Algunos de los problemas

prácticos a los que se enfrentan estas soluciones, radica en la demora en cuanto a procesamiento que puede traer su implementación (Rainer Gerhards, 2010).

2.4. Conclusiones parciales

Se puede concluir que el marco de trabajo contiene los elementos esenciales para la implementación de un sistema de gestión centralizada de trazas de seguridad donde se integren las funciones de generación, búsqueda y visualización de las mismas. Los componentes en la arquitectura propuesta están en concordancia con las principales recomendaciones y normas analizadas. El listado de las herramientas propuestas, soportado sobre herramientas de código abierto, facilita la recolección de los eventos principales a tener en cuenta para la gestión de trazas de seguridad.

Capítulo 3. Aplicación y validación de resultados

En el capítulo siguiente se valida la implementación del marco de trabajo y la arquitectura propuesta en el entorno del nodo de comunicaciones de La Universidad de las Ciencias Informáticas y se presentan los resultados obtenidos.

3.1. Aplicación en el Nodo Central de la Universidad de las Ciencias Informáticas (UCI)

La aplicación del marco de trabajo se realiza mediante la arquitectura que se muestra en el epígrafe 2.2 desplegada parcialmente en el Nodo Central de comunicaciones de la UCI.

El Nodo Central cuenta con más de 150 servidores en producción brindando servicios de correo, navegación hacia Internet, hospedaje de sitios web, mensajería instantánea, entre otros. Se generan más de 30 gigabytes de trazas en formato *gzip* mensualmente lo que representa una gran cantidad de datos a procesar.

En la Figura 3.1 se muestra la integración de la solución en la red del Nodo Central de la UCI donde se ubica el servidor con la instalación de Logstash ELK y el servidor central de Syslog. Los servidores propuestos tienen las características que se muestran en las Tabla 3.1 y Tabla 3.2.

Tabla 3.1. Características del servidor seleccionado para la instalación de Logstash ELK.

Servidor para la instalación de Logstash ELK	
Sistema operativo	CentOS 7
Memoria	8Gbyte
CPU	4 Intel Xeon 2.13 Ghz
Disco	500Gb

Tabla 3.2. Características del servidor seleccionado para la instalación del servidor central de Syslog.

Servidor central de Syslog	
Sistema operativo	CentOS 7
Memoria	8Gbyte
CPU	4 Intel Xeon 2.0 Ghz
Disco	1 Tbyte

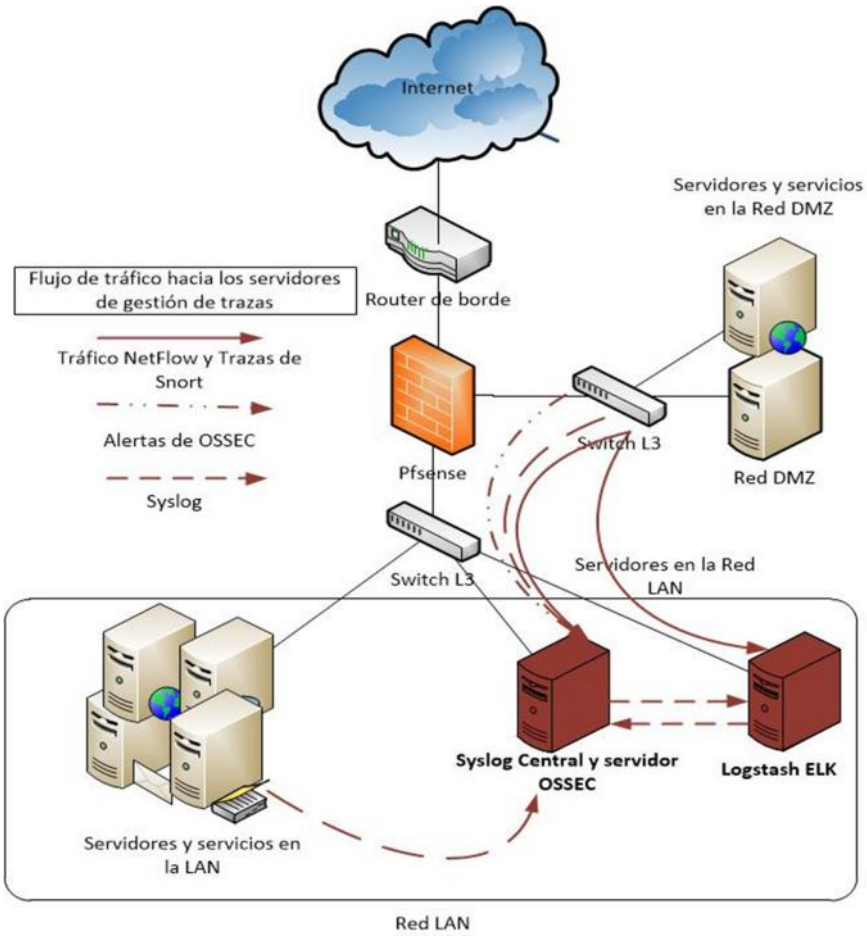


Figura 3.1. Esquema de red del sistema de gestión centralizada de trazas en el Nodo Central de la UCI.

En el servidor Elasticsearch se almacenan las trazas siguientes:

- Trazas generadas en el servidor OSSEC.

- Trazas de flujo de tráfico Netflow.
- Trazas de alertas del IDS Snort.
- Trazas de los servicios de directorio. Controladores de dominio Windows Server 2008.
- Trazas generadas por el sistema operativo de los servidores vía Syslog.

Adicionalmente se puede almacenar cualquier traza de servicio que se recolecte centralmente.

3.2. Análisis de resultados

El proceso de búsqueda en las trazas puede realizarse desde la interfaz web de Kibana o realizando llamadas directamente a la base de datos Elasticsearch. Las operaciones sobre los datos almacenados debe tener mejores tiempos de respuesta que los obtenidos mediante los métodos que normalmente se utilizan con herramientas de búsqueda y filtrado sobre los ficheros almacenados en el disco.

3.2.1. Recolección y almacenamiento de trazas

En el Nodo Central de la UCI se recolectaban cuatro fuentes de trazas asociadas a los servicios de mensajería, Internet y correos interno y externo. Como parte de la implementación del marco de trabajo propuesto, se incluyeron nueve fuentes acorde al análisis realizado en el epígrafe 1.4.

Tabla 3.3. Relación del estado de recolección centralizada de trazas antes y después de implantación de la arquitectura propuesta en la UCI.

Sistemas	Aplicación	Formato	Anteriormente	Estado actual
Navegación proxy web	Squid	Squid	Fuera de línea mediante Rsync	Fuera de línea mediante Rsync
Proxy inverso	Squid	CLF	mediante Rsync	En línea
Relay de correo	Qmail	syslog	Fuera de línea mediante Rsync	En línea
Servicio de correo	Zimbra	syslog	Fuera de línea mediante Rsync	Fuera de línea utilizando Rsync
Mensajería	Openfire	Log4j	Fuera de línea	Fuera de línea

			mediante Rsync	mediante Rsync
Tráfico de red	Sflow	Netflow	No	En línea
Detector de intrusos	Snort	Syslog	No	En línea
Servicio de directorio activo Windows	Directorio activo de Windows 2008, Nxlog	Syslog	No	En línea
Dispositivos de hardware	Software de gestión propietario	Syslog	No	En línea
Sistema operativo Linux	Rsyslog	Syslog	Local en cada servidor	En línea
Sistema operativo Windows	Nxlog	Syslog	No	En línea
Detector de intrusos de *host* OSSEC	OSSEC	Syslog	No	En línea
Cortafuegos	Pfsense	Syslog	No	En Línea

En la Tabla 3.3 se listan el total de trazas recolectadas, incluyendo si se almacenan en línea o fuera de línea. El almacenamiento en línea de las trazas permite el procesamiento instantáneo, posibilitando que puedan detectarse eventos en el periodo de su ocurrencia. En la Figura 3.2 se muestra el resultado de la comparación.

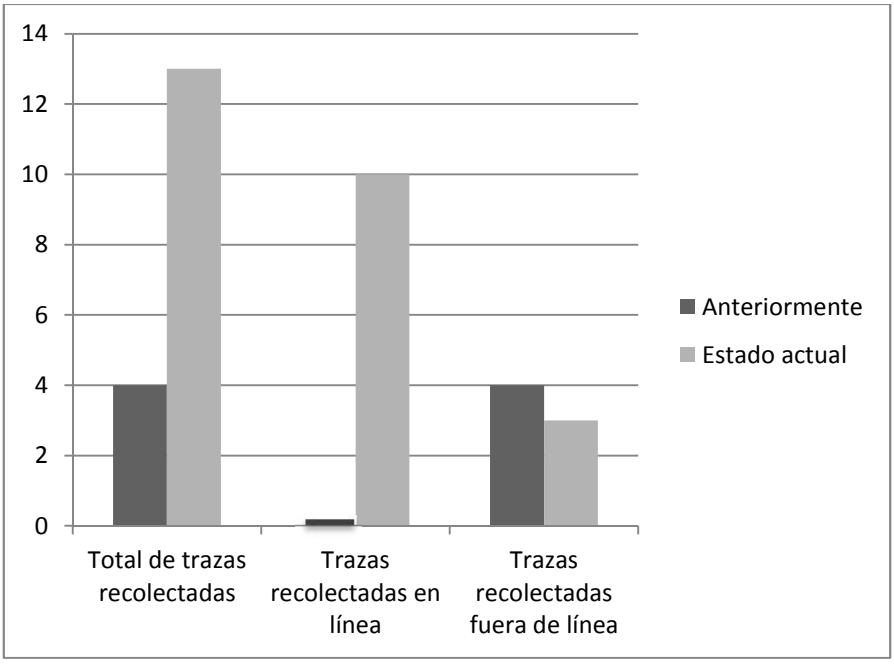


Figura 3.2. Comparación del estado del total de trazas recolectadas antes y después de la aplicación del marco de trabajo.

3.2.2. Agrupación y búsqueda sobre la información almacenada

Desde la interfaz de Kibana se puede agrupar y graficar la información almacenada de las trazas. En la Tabla 3.4 se muestran algunos de los campos que pueden ser agrupados, para obtener los principales registros de eventos asociados. En la búsqueda, se pueden incluir cadenas arbitrarias de texto, términos indexados en combinación con operadores lógicos u otros disponibles en la librería Apache Lucene.

Tabla 3.4. Asociación de los principales eventos que deben ser registrados con la información que puede obtenerse de las trazas almacenadas[37].

Registro de evento	Eventos que pueden ser agrupados
Acciones con privilegios administrativos	EventID: 4672
Acceso a las trazas	Audit: PATH, SYSCALL

[37] No se incluyó ningún registro asociado al monitoreo de fuga de información, ya que no hay una política definida para este tipo de evento al momento de la ejecución de la validación propuesta.

Monitoreo de sesiones	EventID: 4624,4625,4648,4675,4634,4647 Audit: USER_LOGIN, LOGIN, USER_END
Eventos satisfactorios y fallidos de usuarios, aplicaciones y sistemas	EventID: 4624
Creación, modificación y borrado de objetos del sistema	Audit: ADD_GROUP, ADD_USER, DEL_GROUP, DEL_USER EventID: 4738, 4662, 5136
Registro de conexiones remotas	EventID: 6272, 6273
Registro de trazas de tráfico de red	IP fuente, IP destino, puerto fuente, puerto destino, bytes de transferencia
Eventos Críticos	EventID: Severity Syslog: Severity

En la Figura 3.3, se muestra un ejemplo de búsqueda, donde se obtienen los registros asociados a actividades administrativas, en este caso, identificadas por el evento 4672 para el usuario system, utilizando la expresión:

EventID:4672 and system

En el panel superior de Kibana se puede seleccionar el rango de tiempo que se desea cubrir, el panel de la izquierda lista todos los campos identificados en las trazas y el panel principal muestra una gráfica con la cantidad de eventos en el tiempo y la lista de los mismos. En el Anexo 13, se muestran algunos ejemplos de las gráficas obtenidas que resumen la información disponible en las trazas.

De igual forma, la búsqueda realizada puede llevarse a un panel de visualización de forma interactiva y sencilla.

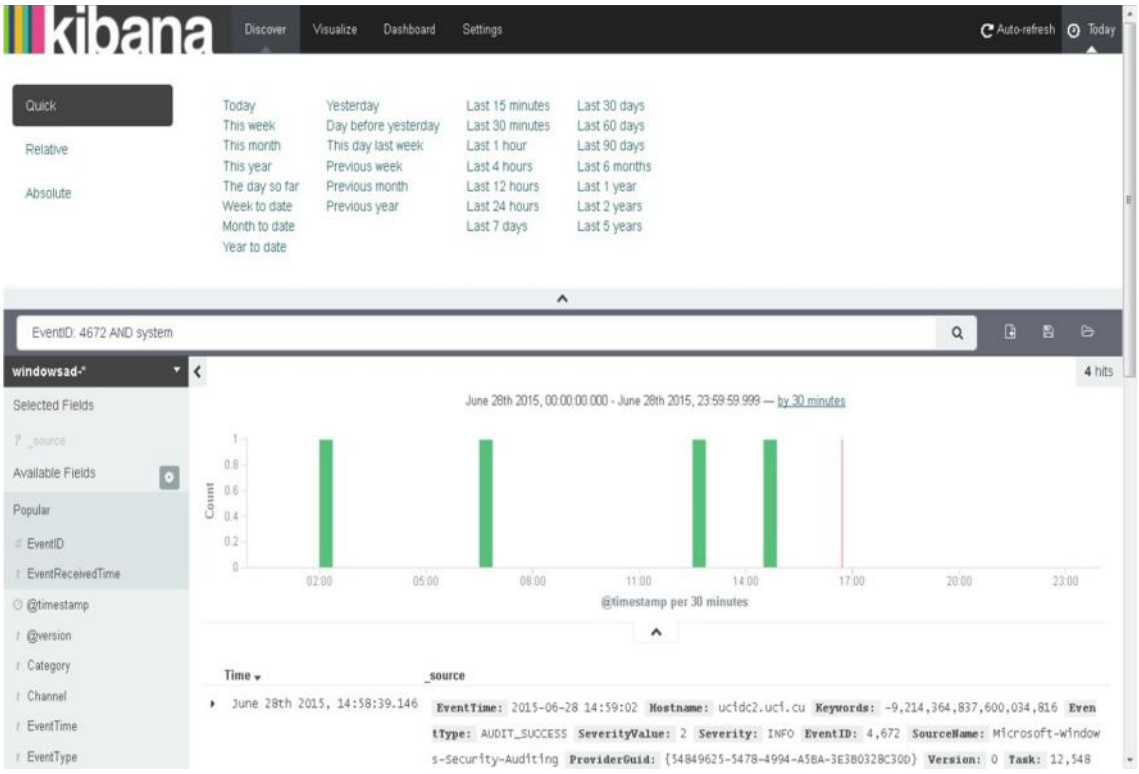


Figura 3.3. Búsqueda de datos en la interfaz de Kibana.

Desde el menú superior de Kibana se selecciona la opción *Visualize*, donde se definen los términos de búsqueda como se muestra en la Figura 3.4.

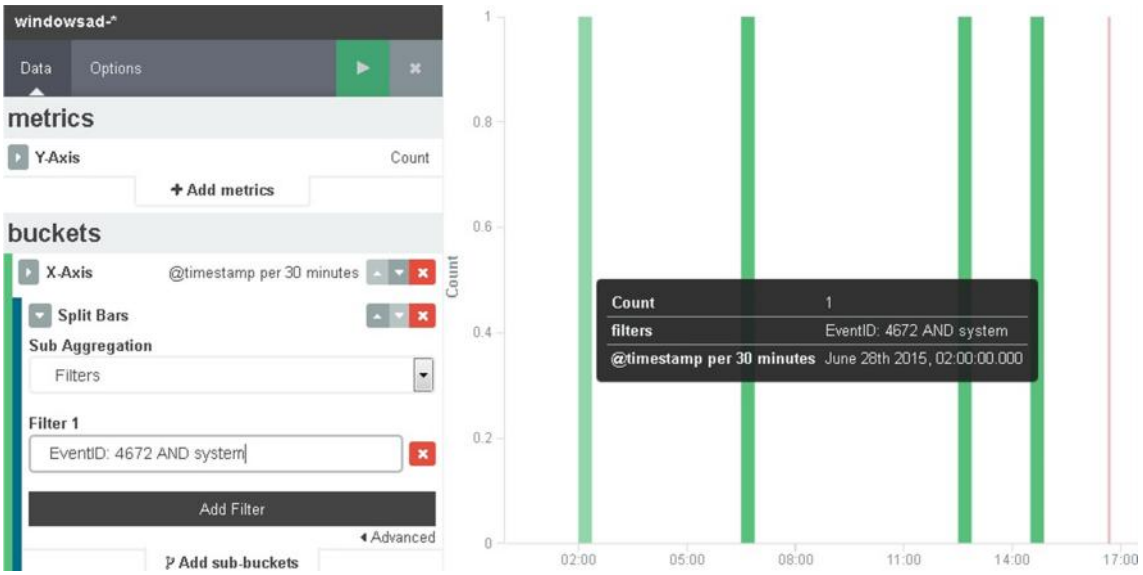


Figura 3.4. Panel para la creación de gráficos en Kibana.

La selección de la búsqueda genera automáticamente la consulta DSL mostrada en la Figura 3.5.

Figura 3.5. Segmento de consulta DSL generada por Kibana.

Aunque el usuario tiene la posibilidad de modificar directamente la consulta mostrada, generalmente esto no es necesario. Kibana permite generar todas las búsquedas y gráficos desde las interfaces destinadas para ello.

Utilizando esta herramienta, se pueden agrupar y relacionar los campos disponibles en cada una de las trazas. También es posible hacer agrupaciones por búsquedas que utilicen filtros y consultas DSL. Una vez hechas las agrupaciones, el especialista de seguridad tiene los datos para conocer cómo se está comportando la red o servicio asociado a las trazas almacenadas.

3.2.3. Análisis de tiempo de búsqueda en las trazas almacenadas

El análisis de trazas utilizando herramientas de consola de Linux requiere de gran experticia de los especialistas. Su utilización se complejiza por múltiples factores como son: Conocer varios estilos de sintaxis, múltiples opciones de línea de comando y el dominio de expresiones regulares. Muchas de las herramientas, como es el caso de *awk*, *sed* o *grep,* pueden utilizarse con propósitos similares o en conjunto. El tiempo para una operación lo determinan: la cantidad de datos a procesar, cómo se use la herramienta y la complejidad de la operación de búsqueda.

Para comprobar el comportamiento del consumo de tiempo con distintas herramientas de consola y haciendo operaciones de búsqueda sobre la base de datos Elasticsearch en la solución propuesta, se plantea el caso de prueba siguiente:

Búsqueda sobre un fichero de trazas equivalente a dos días de tráfico de red netflow almacenadas en el motor Elasticseach y en ficheros en texto plano dividido en varios segmentos. Este tráfico, es de los que más volumen de datos genera en el Nodo Central de la universidad, según el análisis del autor de la investigación, con ficheros de 10 a 12 gigabytes diarios en fichero de texto plano.

La ejecución debe recorrer completamente el fichero para un rango de tiempo mientras agrega el valor del campo *in_bytes* obteniendo la suma total.

Todas las pruebas se realizan sobre una máquina virtual con ocho núcleos Intel *Xeon* a 2.7 GHz y ocho gigabytes de RAM corriendo sobre tecnología para servidores.

La primera búsqueda se realiza comparando distintas variantes de comandos en consola.

Variante 1:

grep -E -o '"in_bytes"\:[[:digit:]]+.[[:digit:]]+' netflow-08-09.log | sed -e 's/"in_bytes"://g' | awk '{s+=$1}END{print s}'

Variante 2:

awk -F, '/in_bytes/{gsub("\"in_bytes\":","",$14);s+=$14}END{print s}' netflow-08-09.log

Variante 3:

awk -F"[,:]" '{s+=$30}END{print s}' netflow-08-09.log

Las tres variantes se aplicaron al fichero *netflow-08-09.log* con un tamaño de 24.96 gigabytes.

La primera, más generalizadora, utiliza una expresión regular para filtrar la cadena *in_bytes* y el valor numérico que tiene a continuación. Pasa el flujo al comando *sed* para obtener el valor numérico y realiza la suma de los valores pasando el flujo a *awk*.

En el segundo caso se utiliza solamente *awk*. Filtrando la cadena *in_bytes* se obtienen los valores numéricos y la suma mediante funciones del propio comando.

En el tercer caso, en *awk,* se obtiene el valor directamente utilizando una expresión regular para el campo configurado.

Tabla 3.5. Tiempo consumido por la combinación de *grep* y *awk* en la búsqueda y procesamiento de un fichero de traza.

Operación de búsqueda	Tiempo consumido
Variante 1	67 minutos y 37 segundos
Variante 2	31 minutos y 9 segundos
Variante 3	7 minutos y 20 segundos

Como se evidencia en la Tabla 3.5, los tiempos varían significativamente en dependencia de la complejidad de los patrones de búsqueda y la utilización de los comandos. En esta prueba, no se ha tenido en cuenta la demora para el diseño del comando a utilizar, que dependerá en gran medida del conocimiento del especialista.

La comparación con la búsqueda sobre Elasticsearch se realizó como sigue:

Para la búsqueda utilizando herramientas UNIX se tuvo en cuenta lo siguiente:

- Se utilizó solamente la herramienta *awk*.
- Se separaron los ficheros en distintas muestras para evitar expresiones regulares de filtrado adicionales.
- En cada ejecución de búsqueda se limpiaron los *buffers* de memoria del sistema operativo.

- Por las características del fichero de trazas, el campo seleccionado no varía en su posición ni estructura lo que permite obtener el valor sin la utilización de expresiones regulares para la selección de un patrón determinado.
- Se ejecutaron cinco corridas cinco corridas con el mismo caso de estudio tomando el valor medio.
- Se tomó como medida el tiempo en milisegundos trascurrido en la ejecución de *awk.*

Para la búsqueda utilizando las funciones que proporciona Elasticsearch se tuvo en cuenta lo siguiente:

- Se utilizó solamente la herramienta *curl*, como herramienta que permite ejecutar llamadas HTTP desde consola para la consulta.
- El motor de Elasticsearch no ejecutó ninguna acción paralela de búsqueda o inserción.
- En cada ejecución de búsqueda se limpiaron los *buffers* de memoria de Elasticsearch y del sistema operativo.
- Se ejecutaron cinco corridas con el mismo caso de estudio tomando el valor medio.
- Se tomó como medida del tiempo de respuesta, el valor proporcionado por el resultado de la búsqueda en Elasticsearch.

Para la comparación del tiempo consumido en cada uno de los ficheros se utilizó en script de *Shell* siguiente:

```
awk -F"[,:]" '{s+=$30}END{print s}
```

Para Elasticsearch se utilizó una cadena de búsqueda equivalente a la siguiente:

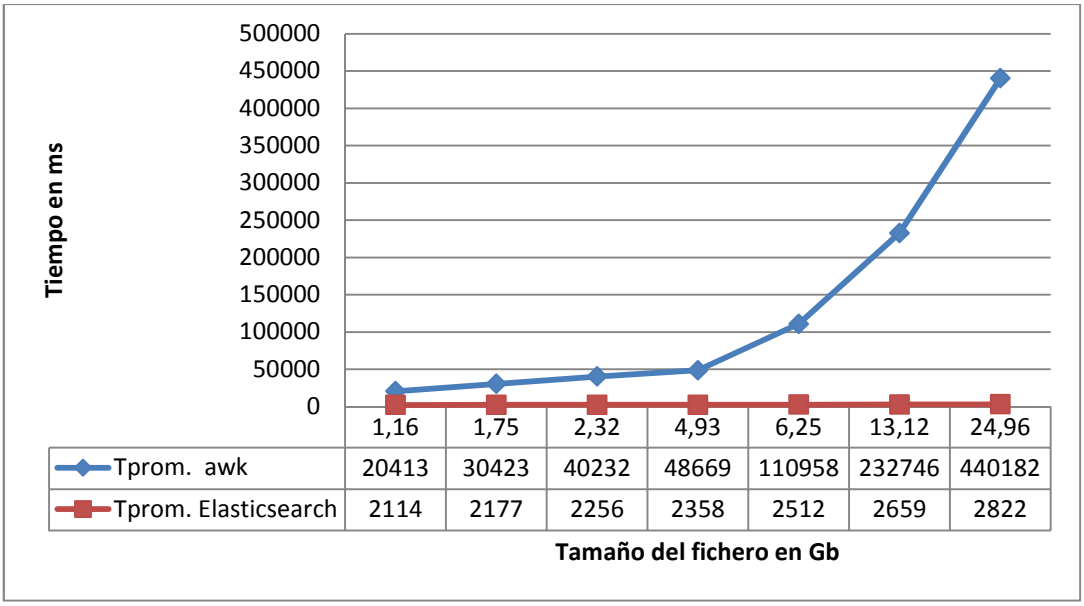
```
{
   "from":0,
   "size":0,
   "query":{
      "range":{
         "@timestamp":{
            "gte":"2015-05-09T00:00:00",
            "lte":"2015-05-09T00:00:00"
         }
      }
   },
```

```
   "aggs":{
      "in_bytes_return":{
         "sum":{
            "field":"in_bytes"
         }
      }
   }
}
```

En la Tabla 3.6 se muestran los resultados obtenidos para cada caso de prueba y en la Figura 3.6 la comparación entre los tiempos medios consumidos por cada fichero de traza.

Tabla 3.6.Tiempo de búsqueda medio utilizando la herramienta *awk* y Elasticsearch para muestras de ficheros de trazas de distintos tamaños.

Tamaño en gigabytes	Cantidad de líneas	Tiempo en ms medio con awk	Tiempo medio en ms con Elasticseach
1,16	1.559.241	20413	2114
1,75	2.352.450	30423	2177
2,32	2.352.450	40232	2256
4,93	3.811.720	48669	2358
6.25	16.407.627	110958	2512
13,12	16.407.627	232746	2659
24,96	33.655.533	440182	2822



	1,16	1,75	2,32	4,93	6,25	13,12	24,96
Tprom. awk	20413	30423	40232	48669	110958	232746	440182
Tprom. Elasticsearch	2114	2177	2256	2358	2512	2659	2822

Figura 3.6. Comparación de los tiempos de respuesta promedio en la búsqueda de un patrón determinado en distintos ficheros de trazas utilizando herramientas Unix y Elasticsearch.

Como se puede ver en la Figura 3.6, los tiempos de respuesta en Elasticsearch se mantienen con valores estables sobre los dos segundos. Este es un comportamiento significativamente menor que los resultados obtenidos utilizando herramientas de búsqueda de patrones en ficheros de texto. En el caso de Elasticsearch, no se ha ejecutado ningún tipo de optimización del indexado o de la aplicación. Modificaciones en estos parámetros pueden mejorar los tiempos de respuesta obtenidos.

3.3. Conclusiones parciales

En este capítulo se presentaron los resultados obtenidos partiendo de la aplicación del marco de trabajo en el Nodo Central de la UCI. Se incluyeron nuevas fuentes de trazas a partir de la investigación realizada. Se Implementó la arquitectura de gestión de trazas de seguridad propuesta, mostrándose las búsquedas en los distintos paneles de visualización, facilitando la obtención de los resultados. Se evidenció en mediciones realizadas, la rapidez de las búsquedas en grandes volúmenes de datos, manteniendo un tiempo aproximadamente constante, contra la utilización de las herramientas clásicas de búsqueda, donde la demora aumentaba casi exponencialmente con el aumento de la cantidad de datos.

CONCLUSIONES

El proceso de gestión de trazas de seguridad requiere del análisis de grandes volúmenes de datos generados por múltiples fuentes lo que provoca lentitud en el procesamiento, búsqueda y visualización de la información contenida en las mismas.

Del análisis de las principales regulaciones y normas existentes se obtuvieron, mediante un proceso de selección, los componentes de un sistema de gestión de trazas y se realizó una síntesis que recoge los eventos de mayor importancia para procesar y analizar.

Se desarrolló un marco de trabajo para la gestión centralizada de trazas de seguridad, que incluye la selección de los eventos más importantes descritos en las principales regulaciones y normas. Se seleccionaron las herramientas de código abierto para la recolección de estos eventos, siguiendo un enfoque integrador que extiende las propuestas existentes.

El marco de trabajo propuesto fue implementado en el nodo de comunicaciones de la UCI, donde se incorporaron nuevas fuentes de trazas al sistema de gestión existente. Como parte del proceso de validación se pudo comprobar que se facilitó el procesamiento, y se disminuyó el tiempo de búsqueda y visualización de las trazas mediante el almacenamiento estructurado y la utilización de las funcionalidades brindadas por la integración de las herramientas de código abierto seleccionadas.

RECOMENDACIONES

A partir de la investigación realizada se proponen las siguientes recomendaciones:

- Proponer métricas de seguridad que puedan incluirse en los paneles de visualización de Kibana a partir de la información procesada, añadiéndole valor a la información obtenida de las trazas.
- Desarrollar una aplicación que posibilite realizar acciones que generen alertas en base a métricas preestablecidas leyendo los datos que se almacenan en Elasticseach.
- Desarrollar paneles de mando genéricos para Kibana que puedan ser utilizados en otros entornos donde se aplique el marco de trabajo propuesto.
- Añadir las configuraciones necesarias para la extensión de la arquitectura desarrollada, en un entorno distribuido, con más de un servidor Elasticsearch.

REFERENCIAS BIBLIOGRÁFICAS

ACCELOPS 2013. Good Practice Guide (GPG13) Compliance in the UK. Successful Protective Monitoring with AccelOps. [en línea]. S.l.: [Consulta: 9 marzo 2015]. Disponible en: http://www.accelops.com/media/1697/AccelOpsGoodPracticeGuideSolutionBrief.pdf.

ALBING, C., VOSSEN, J.P. y NEWHAM, C. 2007. *bash Cookbook: Solutions and Examples for bash Users*. S.l.: O'Reilly Media, Inc. ISBN 0-596-55470-2.

ANTON CHUVAKIN, 2010. *The Complete Guide to Log and Event Management* [en línea]. marzo 2010. S.l.: s.n. [Consulta: 23 marzo 2013]. Disponible en: http://www.novell.com/docrep/2010/03/Log_Event_Mgmt_WP_DrAntonChuvakin_March2010_Single_en.pdf.

A. OKMIANSKI 2009. RFC 5426 - Transmission of Syslog Messages over UDP. [en línea]. [Consulta: 26 abril 2013]. Disponible en: http://tools.ietf.org/html/rfc5426.

ARTYOM CHURILIN 2013. *CHOOSING AN OPEN-SOURCE LOG MANAGEMENT SYSTEM FOR SMALL BUSINESS* [en línea]. S.l.: TALLINN UNIVERSITY OF TECHNOLOGY. [Consulta: 1 mayo 2015]. Disponible en: http://lab.cs.ttu.ee/dl135.

BAKER, A.R., ESLER, J. y ALDER, R. 2007. Snort IDS and IPS toolkit. *Syngress, Canada*,

BELLARE, M. y YEE, B. 1997. Forward integrity for secure audit logs. . S.l.: Citeseer.

BOTOND BOTYANSZKI 2009a. NXLOG Community Edition Reference Manual for v2.8.1248 | nxlog.co. [en línea]. [Consulta: 21 enero 2015]. Disponible en: http://nxlog.org/documentation/nxlog-community-edition-reference-manual-v20928.

BOTOND BOTYANSZKI 2009b. NXLOG Community Edition Reference Manual for v2.9.1357. [en línea]. [Consulta: 22 junio 2015]. Disponible en: http://107.170.5.221/docs/nxlog-ce/nxlog-reference-manual.html.

BRAY, R., CID, D. y HAY, A. 2008a. *OSSEC host-based intrusion detection guide*. S.l.: Syngress. ISBN 0080558771.

BRAY, R., CID, D. y HAY, A. 2008b. *OSSEC host-based intrusion detection guide*. S.l.: Syngress. ISBN 0-08-055877-1.

BUTLER, J.M. 2009. *Benchmarking security information event management*. S.l.: Tech. rep., SANS, 2 2009.

CARVEY, H. 2014. *Windows Forensic Analysis Toolkit: Advanced Analysis Techniques for Windows 8*. S.l.: Elsevier. ISBN 0-12-417174-5.

CASWELL, B., BEALE, J. y BAKER, A. 2007. *Snort Intrusion Detection and Prevention Toolkit*. S.l.: Syngress. ISBN 0-08-054927-6.

CHUVAKIN, A., SCHMIDT, K. y PHILLIPS, C. 2012. *Logging and log management: the authoritative guide to understanding the concepts surrounding logging and log management*. S.l.: Newnes. ISBN 1-59749-636-7.

CISCO SYSTEMS, INC., F.M., Ed., Y. Ma, Ed., Huawei Technologies,J. Salowey, Ed. 2009. RFC 5425 - Transport Layer Security (TLS) Transport Mapping for Syslog. [en línea]. [Consulta: 27 abril 2013]. Disponible en: http://tools.ietf.org/html/rfc5425.

C. LONVICK 2001. RFC 3164 - The BSD Syslog Protocol. [en línea]. [Consulta: 26 abril 2013]. Disponible en: http://tools.ietf.org/html/rfc3164.

COOPER, M. 2014. *Advanced bash-scripting guide*. S.l.: s.n. ISBN 1-146-06872-7.

COUNCIL, P.C.I. 2010. *PCI DSS 2.0*. S.l.: PCI Council Publication/United States.

CSIS 2013. CSIS: 20 Critical Security Controls. [en línea]. [Consulta: 28 abril 2013]. Disponible en: http://www.sans.org/critical-security-controls/.

DAVE SHACKLEFORD, 2008. *Leveraging Event and Log Data for Security and Compliance*. abril 2008. S.l.: SANS.

ELASTICSEARCH 2015. Kibana User Guide [4.1]. [en línea]. [Consulta: 22 junio 2015]. Disponible en: https://www.elastic.co/guide/en/kibana/current/index.html.

FITZGERALD, E. y HEINBOCKEL, B. 2010. COMMON EVENT EXPRESSION (CEE) OVERVIEW. ,

F. MIAO 2009. RFC 5425 - Transport Layer Security (TLS) Transport Mapping for Syslog. [en línea]. [Consulta: 27 abril 2013]. Disponible en: http://tools.ietf.org/html/rfc5425.

FRY, C. y NYSTROM, M. 2009. *Security monitoring*. S.l.: O'Reilly Media, Inc. ISBN 0-596-55545-8.

F. YERGEAU 2003. RFC 3629. *UTF-8, a transformation format of ISO 10646* [en línea]. [Consulta: 7 mayo 2015]. Disponible en: http://tools.ietf.org/html/rfc3629.

GERHARDS, R. 2010. Rsyslog: going up from 40K messages per second to 250K. *Linux Kongress*. S.l.: s.n.,

GORMLEY, C. y TONG, Z. 2014. *Elasticsearch: The Definitive Guide*. S.l.: O'Reilly & Associates. ISBN 1-4493-5854-3.

GUTTMAN, B. y ROBACK, E.A. 1995. SP 800-12. *An Introduction to Computer Security: the NIST Handbook, National Institute of Standards & Technology, Gaithersburg, MD*,

ISO, I.E.C. 2005. ISO 27002: 2005. *Information Technology-Security Techniques-Code of Practice for Information Security Management. ISO*,

JAYANT KUMAR 2015. *Apache Solr Search Patterns*. S.l.: Packt Publishing. ISBN 978-1-78398-184-7.

JERRY SHENK, 2012. *SANS Eighth Annual 2012 Log and Event Management Survey Results* [en línea]. mayo 2012. S.l.: SANS. [Consulta: 23 marzo 2013]. Disponible en: http://www.sans.org/reading_room/analysts_program/SortingThruNoise.pdf.

JERRY SHENK, 2014. *Ninth Log Management Survey Report* [en línea]. octubre 2014. S.l.: SANS. [Consulta: 25 junio 2015]. Disponible en: http://www.sans.org/reading-room/whitepapers/analyst/ninth-log-management-survey-report-35497.

J. KELSEY 2010. RFC 5848 - Signed Syslog Messages. [en línea]. [Consulta: 27 abril 2013]. Disponible en: http://tools.ietf.org/html/rfc5848.

JORDAN SISSEL 2012. experiments/elasticsearch/disk at master · jordansissel/experiments · GitHub. [en línea]. [Consulta: 14 mayo 2015]. Disponible en: https://github.com/jordansissel/experiments/tree/master/elasticsearch/disk.

KÄLLQVIST, E. y LAM, J.Q. 2012. Reliable and Tamper Resistant Centralized Logging in a High Availability System-An Investigation on Ericsson SGSN-MME. ,

KEMP, J. 2009. *Linux System Administration Recipes: A Problem-solution Approach*. S.l.: Apress. ISBN 1-4302-2450-9.

KENT, K. y SOUPPAYA, M.P. 2006. SP 800-92. *Guide to Computer Security Log Management, National Institute of Standards & Technology, Gaithersburg, MD*,

KUĆ, R. y ROGOZIŃSKI, M. 2013. *Mastering ElasticSearch*. S.l.: Packt Publishing Ltd. ISBN 1-78328-144-8.

MA, D. y TSUDIK, G. 2009. A new approach to secure logging. *ACM Transactions on Storage (TOS)*, vol. 5, no. 1, pp. 2.

MIC, 2007. *Resolución 127/2007 MIC. Reglamento de seguridad para las tecnologías de la información. Ministerio de la informática y las comunicaciones (MIC)*. 2007. S.l.: s.n.

MICHAEL KERRISK 2014. utmp(5) - Linux manual page. [en línea]. [Consulta: 13 enero 2015]. Disponible en: http://man7.org/linux/man-pages/man5/utmp.5.html.

MICHAEL, M., ERIK, H. y OTIS, G. 2010. *Lucene in Action*. S.l.: Stamford: Manning Publications Co.

MICLEA, S. 2012. Windows and Linux Security Audit. *Journal of Applied Business Information Systems*, vol. 3, no. 4, pp. 117.

MICROSOFT 2009. Event Log. [en línea]. [Consulta: 13 enero 2015]. Disponible en: http://technet.microsoft.com/en-us/library/dd349798(v=ws.10).aspx.

MICROSOFT 2015a. EVENTLOGRECORD structure. *Windows Dev Center* [en línea]. [Consulta: 13 enero 2015]. Disponible en: http://msdn.microsoft.com/en-us/library/windows/desktop/aa363646(v=vs.85).aspx.

MICROSOFT 2015b. Event Types (Windows). *Windows Dev Center* [en línea]. [Consulta: 13 enero 2015]. Disponible en: http://msdn.microsoft.com/en-us/library/windows/desktop/aa363662(v=vs.85).aspx.

MILLER, D. y PEARSON, B. 2011. *Security information and event management (SIEM) implementation*. S.l.: McGraw-Hill Companies. ISBN 0-07-170109-5.

M. ROSE 2001. RFC 3195 - Reliable Delivery for syslog. [en línea]. [Consulta: 26 abril 2013]. Disponible en: http://tools.ietf.org/html/rfc3195.

NIST, S. 2007. 800-53. *Recommended Security Controls for Federal Information Systems*, pp. 800-53.

RAINER GERHARDS 2008. Rainer's Blog: On the (un)reliability of plain tcp syslog... [en línea]. [Consulta: 1 mayo 2015]. Disponible en: http://blog.gerhards.net/2008/04/on-unreliability-of-plain-tcp-syslog.html.

RAINER GERHARDS, 2010. *CEE Log Integrity and the \Counterpane Paper" [1]* [en línea]. 12 febrero 2010. S.l.: s.n. [Consulta: 13 marzo 2015]. Disponible en: http://www.gerhards.net/download/log_hash_chaining.pdf.

RAINER GERHARDS 2014a. RELP - The Reliable Event Logging Protocol (Specification). [en línea]. [Consulta: 1 mayo 2015]. Disponible en: http://www.rsyslog.com/doc/relp.html.

RAINER GERHARDS 2014b. RELP - The Reliable Event Logging Protocol (Specification). [en línea]. [Consulta: 29 enero 2015]. Disponible en: http://www.rsyslog.com/doc/relp.html.

R. GERHARDS 2009. RFC 5424 - The Syslog Protocol. [en línea]. [Consulta: 28 abril 2015]. Disponible en: https://tools.ietf.org/html/rfc5424.

RITTINGHOUSE, J.W. y RANSOME, J.F. 2009. *Cloud computing: implementation, management, and security*. S.l.: CRC press. ISBN 1-4398-0681-0.

ROBERT KRÁTKÝ, M.S., YOANA RUSEVA, S.W. y TOMÁŠ ČAPEK, M.P. 2015. Security Guide A Guide to Securing Red Hat Enterprise Linux 7. [en línea]. [Consulta: 23 junio 2015]. Disponible en: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Security_Guide/index.html.

RYBACZYK, P. 2005. *Expert Network Time Protocol: An Experience in Time with NTP*. S.l.: Apress. ISBN 1-4302-0039-1.

SAWANT, N. y SHAH, H. 2013. *Big Data Application Architecture Q&A: A Problem-Solution Approach*. S.l.: Apress. ISBN 1-4302-6293-1.

SCHNEIER, B. y KELSEY, J. 1998. Cryptographic Support for Secure Logs on Untrusted Machines. *USENIX Security*. S.l.: s.n.,

SKIBIŃSKI, P. y SWACHA, J. 2007. Fast and efficient log file compression. *CEUR Workshop Proceedings of the 11th East-European Conference on Advances in Databases and Information Systems (ADBIS)*. S.l.: s.n., pp. 330-342.

SOBELL, M.G. 2005. *A practical guide to Linux commands, editors, and shell programming*. S.l.: Prentice Hall Professional Technical Reference. ISBN 0-13-147823-0.

THE MITRE CORPORATION, 2010. *Common Event Expression, Architecture Overview, Version 0.5*. mayo 2010. S.l.: The CEE Editorial Board.

TURNBULL, J. 2013. *The Logstash Book*. S.l.: James Turnbull. ISBN 0-9888202-1-8.

ANEXOS

Anexo 1. Formato de mensajes del protocolo Syslog

Formato de mensajes del protocolo Syslog

```
SYSLOG-MSG      = HEADER SP STRUCTURED-DATA [SP MSG]

      HEADER          = PRI VERSION SP TIMESTAMP SP HOSTNAME
                        SP APP-NAME SP PROCID SP MSGID
      PRI             = "<" PRIVAL ">"
      PRIVAL          = 1*3DIGIT ; range 0 .. 191
      VERSION         = NONZERO-DIGIT 0*2DIGIT
      HOSTNAME        = NILVALUE / 1*255PRINTUSASCII

      APP-NAME        = NILVALUE / 1*48PRINTUSASCII
      PROCID          = NILVALUE / 1*128PRINTUSASCII
      MSGID           = NILVALUE / 1*32PRINTUSASCII

      TIMESTAMP       = NILVALUE / FULL-DATE "T" FULL-TIME
      FULL-DATE           =  DATE-FULLYEAR  "-"  DATE-MONTH  "-"
DATE-MDAY
      DATE-FULLYEAR   = 4DIGIT
      DATE-MONTH      = 2DIGIT  ; 01-12
      DATE-MDAY        = 2DIGIT  ; 01-28, 01-29, 01-30, 01-31
based on
                               ; month/year
      FULL-TIME       = PARTIAL-TIME TIME-OFFSET
      PARTIAL-TIME     = TIME-HOUR ":" TIME-MINUTE ":" TIME-
SECOND
                        [TIME-SECFRAC]
      TIME-HOUR       = 2DIGIT  ; 00-23
      TIME-MINUTE     = 2DIGIT  ; 00-59
      TIME-SECOND     = 2DIGIT  ; 00-59
      TIME-SECFRAC    = "." 1*6DIGIT
      TIME-OFFSET     = "Z" / TIME-NUMOFFSET
      TIME-NUMOFFSET  = ("+" / "-") TIME-HOUR ":" TIME-MINUTE


      STRUCTURED-DATA = NILVALUE / 1*SD-ELEMENT
      SD-ELEMENT      = "[" SD-ID *(SP SD-PARAM) "]"
      SD-PARAM        = PARAM-NAME "=" %d34 PARAM-VALUE %d34
      SD-ID           = SD-NAME
      PARAM-NAME      = SD-NAME
      PARAM-VALUE       = UTF-8-STRING ; characters '"', '\'
and
                                  ; ']' MUST be escaped.
      SD-NAME         = 1*32PRINTUSASCII
                        ; except '=', SP, ']', %d34 (")
```

```
      MSG             = MSG-ANY / MSG-UTF8
      MSG-ANY         = *OCTET ; not starting with BOM
      MSG-UTF8        = BOM UTF-8-STRING
      BOM             = %xEF.BB.BF


Gerhards                                        Standards  Track
[Page 8]
      UTF-8-STRING    = *OCTET ; UTF-8 string as specified
                        ; in RFC 3629

      OCTET           = %d00-255
      SP              = %d32
      PRINTUSASCII    = %d33-126
      NONZERO-DIGIT   = %d49-57
      DIGIT           = %d48 / NONZERO-DIGIT
      NILVALUE        = "-"
```

Anexo 2. Códigos asociados a los recursos (*facility*) en los mensajes Syslog

Código numérico	Recursos
0	Mensajes de kernel
1	Mensajes de nivel de usuario
2	Sistema de correo
3	Demonios del sistema
4	Mensajes de seguridad /Autorización
5	Mensajes generados internamente por syslog
6	Subsistema de impresión
7	Sistema de noticias de red
8	Subsistema UUCP
9	Demonio de reloj
10	Mensajes de seguridad /Autorización
11	Demonio FTP
12	Subsistema NTP
13	Trazas de auditoría
14	Trazas de alerta
15	Demonio de reloj

16	Local 0
17	Local 1
18	Local 2
19	Local 3
20	Local 4
21	Local 5
22	Local 6
23	Local 7

Anexo 3. Niveles de Importancia (*severity*) de los mensajes Syslog

Código numérico	Importancia
0	Emergency: El sistema no se puede utilizar
1	Alert: Se debe tomar una acción de forma inmediata
2	Critical: Condición crítica
3	Error: Condición de error
4	Warning: Condición de aviso
5	Notice: Normal, condición significante
6	Informational: Mensaje de información
7	Debug: Mensaje de depuración

Anexo 4. Principales características de Syslog- NG

Característica	Descripción
Licencia	Licenciamiento dual, Versión de código fuente abierto(Syslog NG CE) y versiones de pago(Syslog NG PE, Syslog NG Store Box)
Plataformas	Linux, BSDs, Solaris, HP-UX, AIX, Tru64

Fuente de eventos	RFC3164, UDP RFC5424
Transportes	Permite recibir mensajes de: IPv6, UNIX domain socket, UDP, UDP RFC5424, TCP, TLS RFC5424, tuberías, ficheros. Permite enviar mensajes a: IPv6, UNIX domain socket, UDP, TCP, soporte TLS.
Soporte para bases de datos	MySQL, Microsoft SQL (MSSQL), Oracle, PostgreSQL, and SQLite.
Otras características avanzadas	
Características solo disponibles en la versión empresarial	Agente syslog-ng disponible para Windows, soporte nativo para bases de datos, soporte para RLTP (Reliable Log Transfer Protocol), copia de mensajes a una cola en disco, mensajes en crudo que no siguen el formato Syslog, múltiples opciones de filtrado por cada uno de los campos de los mensajes, expresiones condicionales y expresiones regulares.

Anexo 5. Características fundamentales de Rsyslog

Características	**Descripción**
Licencia	GPLv3 (GPLv2 para las versiones 2.x)
Plataformas	Linux/BSD
Fuente de eventos	Unix domain sockets, UDP, TCP, Reliable logging library(RELP), kernel log, ficheros
Transportes	Soporte para IPv6, Syslog sobre TCP, UDP Syslog, Syslog sobre RELP, recepción de mensajes según RFC3195, Envio de traps SNMP, mensajes Syslog sobre TLS/SSL, envío a MongoDB, envío a ElasticSearch
Soporte para bases de datos	Soporte nativo(MySQL,PostgreSQL), módulo molibdbi(Oracle, SQLite, Microsoft SQL, Sybase,

	Firebird/Interbase, Ingres, mSQL)
Otras características avanzadas	Compatibilidad con syslogd, compresión de mensajes vía Zlib en la transmisión, copia de mensajes a una cola en disco bajo demanda, formato de mensajes personalizado mediante la utilización de plantillas(CSV, JSON), múltiples opciones de filtrado por cada uno de los campos de los mensajes, expresiones condicionales y expresiones regulares

Anexo 6. Características principales de Nxlog

Características	**Descripción**
Licencia	GPL/LGPL (Nxlog CE) con una versión empresarial de pago(Nxlog EE)
Plataformas	Linux (Debian, Redhat, Ubuntu), BSD, HPUX, Android y Microsoft Windows
Fuente de eventos	TCP, UDP, TLS/SSL, Unix Domain Socket
Transportes	TCP, UDP, TLS/SSL, Unix Domain Socket
Soporte para bases de datos	Lectura y copia de trazas desde PostgreSQL, MySQL, Oracle, MsSQL, SqlLite, Sybase
Otras características avanzadas	Formato de mensajes según RFC 3164 and RFC 5424), CSV, JSON, XML, GELF, Windows EventLog, Procesamiento de mensajes fuera de línea, copia de mensajes a una cola en disco según condiciones de carga priorización. Múltiples opciones de filtrado por cada uno de los campos de los mensajes, expresiones condicionales y expresiones regulares
Características solo disponibles en la versión empresarial	Permite recibir comandos SNMP, gestión remota de los agentes, recolección remota de eventos de Windows mediante WMI. Envío y recepción de mensajes vía HTTP/S, interface ODBC para la recepción y copia de trazas, soporte de un servidor externo de tiempo.

Anexo 7. Aplicación para el análisis de un fichero de trazas

La aplicación, desarrollada en *Perl*, lee un fichero de trazas en texto claro o comprimido en formato *gzip* donde cada evento ocupe una sola línea y devuelve:

- Total de líneas en el fichero.
- Total de caracteres.
- Promedio de caracteres por línea.
- Mayor cantidad de caracteres en una línea.
- Tamaño en bytes y megabytes del fichero de trazas.
- Tiempo de demora de ejecución en segundos des script.

Cálculo de parámetros de un fichero de trazas

```
use File::stat;
use Time::HiRes;
my $count1=0;
my $count2=0;
my $totalLines=0;
my $maxLine=0;
my ($file) = @ARGV;
my $statf = stat($file);
my $start = Time::HiRes::gettimeofday();
if ($file =~ /\.gz$/) {
  open($FILED, "gunzip -c $file |") || die "Error: $!";
}
else{
  open ($FILED, $file) or die "Error: $!";
}
my %result = ();
while(<$FILED>){
        $totalLines = $totalLines + 1;
        $lenLine =  length($_);
        if ($maxLine < $lenLine){
```

```
            $maxLine = $lenLine;
        }
        $cnt =  $cnt + $lenLine;
        $count1++;
        if ($count1 == 1000000 ){
            $count2++;
            print $count2,"x1M líneas procesadas.\n";
            $count1=0;
        }
    }
    my $end = Time::HiRes::gettimeofday();
    printf("Procesado en: %.4f s\n", $end - $start);
    #print "-----------------------------------------------------\n";
    close $FILED;
    while (my ($k,$v)=each %result){print "$k $v\n"}
    print $totalLines, " :Total de líneas","\n";
    print $cnt," :Total de caracteres","\n";
    printf "%.2f %s %s",$cnt/$totalLines,":Promedio de caracteres por línea:","\n";
    print $maxLine," :Mayor cantidad de caracteres en una línea:","\n";
printf  "%d %s %d %.2f %s %s", $statf->size,":Tamaño en bytes" , $statf->size,$statf->size/(1024*1024),"Mbytes" , "\n";
```

Anexo 8. Cálculo de la tasa de compresión para una muestra de ficheros de trazas

Para el cálculo se utilizaron los datos siguientes:

- 29 registros diarios de ficheros de trazas asociadas a las aplicaciones Openfire, servidor de correo Qmail, trazas de navegación web con Squid proxy y trazas de los accesos a la interfaz web de un servidor Nginx.

  El porciento de compresión va a estar dado por la relación:

$$\% Cp = \frac{Fichero\ comprimido}{Fichero\ descomprimido}\ x\ 100$$

  Todos los tamaños son en kilobytes

Tipo de traza	Gzip	Texto plano	%Cp. Min.	%Cp. Máx.	%Cp prom.
Openfire	785,52	7918,76	9.55	11.08	10,05
Qmail	18474,62	109723,86	14.82	18.42	16,84
Squid	99635,17	573978,90	15.78	18.89	17,18
Nginx	13060,00	186352,00	5,84	7,14	7,01

Anexo 9. Configuración básica de Nxlog en Windows para la recolección de eventos del sistema y desde un fichero de trazas en formato texto

C:\Program Files (x86)\nxlog\conf\nxlog.conf

```
<Input file_watch>
Module im_file
File "C:\WINDOWS\system32\LogFiles\IN*.log"
SavePos TRUE
</Input>
<Input mseventlog>
Module im_msvistalog
Query <QueryList>\
 <Query Id="0">\
<Select Path="Security">*</Select>\
<Select Path="Application">*</Select>    \
<Select Path="System">*</Select>    \
 </Query>\
 </QueryList>
 Exec to_json();
</Input>
<Output out>
  Module om_tcp
  Host 1.2.3.4
  Port 1514
```

```
</Output>
<Route 1>
Path internal, file_watch, mseventlog => out
</Route>
```

Anexo 10. Aplicación para la compresión y generación de función resumen de los ficheros de trazas almacenados centralmente

```
#!/usr/bin/python
#https://www.loggly.com/docs/python-syslog/
import sys
import os
import glob
import logging
import logging.handlers
class Logcompress():
    def __init__(self):
        pass
    def Logger(self,msg):
        logging.basicConfig(level=logging.INFO)
        self.logger = logging.getLogger(__name__)
        self.handler = logging.handlers.SysLogHandler('/dev/log')
        self.logger.addHandler(self.handler)
        self.formatter = logging.Formatter('Python: { "loggerName":"%(name)s",
"asciTime":"%(asctime)s", "pathName":"%(pathname)s",
"logRecordCreationTime":"%(created)f", "functionName":"%(funcName)s",
"levelNo":"%(levelno)s", "lineNo":"%(lineno)d", "time":"%(msecs)d",
"levelName":"%(levelname)s", "message":"%(message)s"}')
        self.handler.formatter = self.formatter
        self.logger.info(msg)
    def compress_file(self,infilename):
```

```
        if glob.glob(infilename):
            status = os.system("find "  + infilename + " -type f -mtime +0 -exec gzip -9 {} \; -
exec sha256sum " + infilename + ".gz >  "  + infilename + ".log.gz.sign "+ "  \;")
            if status != 0:
                self.Logger("Error executing external command find or sha256sum")
if __name__ == "__main__":
    log = Logcompress()
    try:
        directory=sys.argv[1]
        for root, subdirs, files in os.walk(directory):
            for i in files:
                filec = root +'/' + i.split(".")[0] + ".log"
                log.compress_file(filec)
    except Exception as e:
            Logger(str(e))
```

Anexo 11. Configuración de Logstash

Fichero de configuración de Logstash para la recolección y almacenamiento de trazas generadas por Nxlog vía TCP.

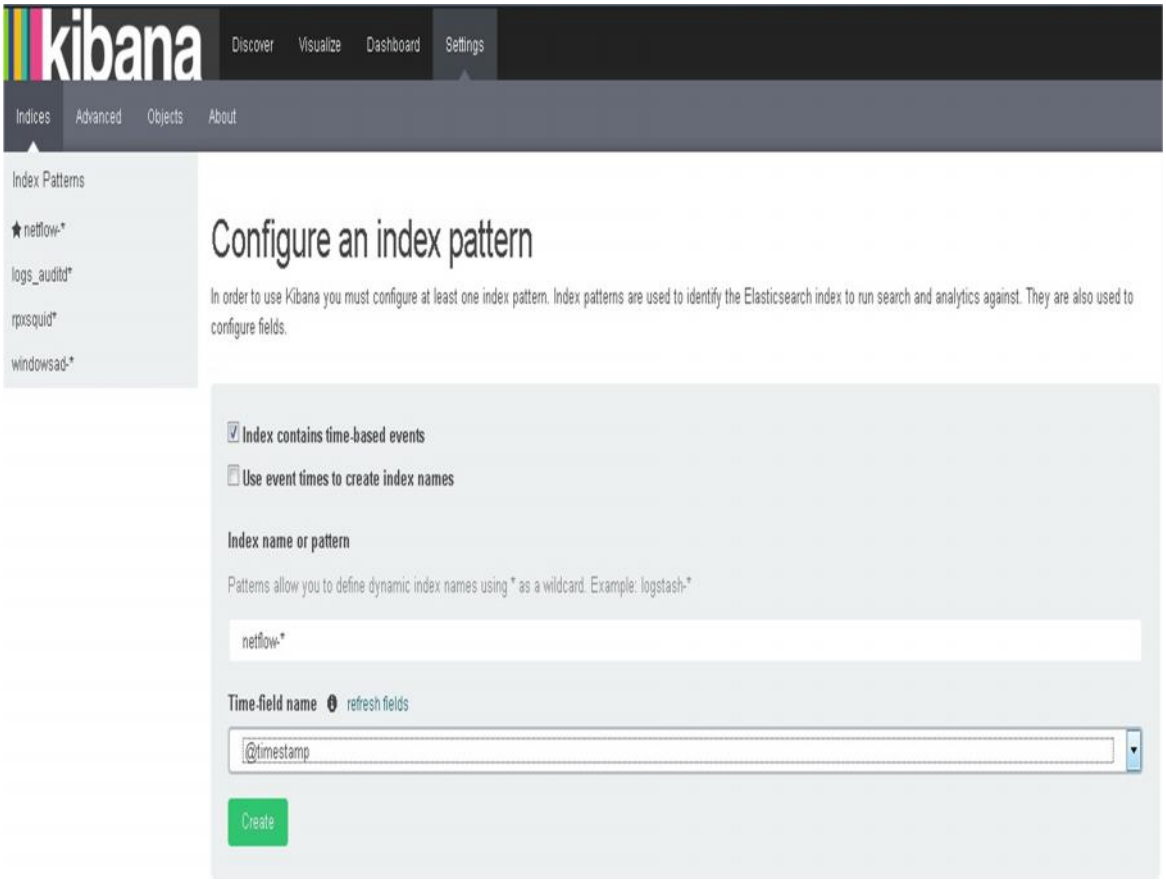
```
input {
        tcp {
          port => 2514
          codec => json_lines
          type => nxlogjson
          tags => ['ucidc']
        }
}

input {
        tcp {
          port => 2515
          codec => json_lines
          type => nxlogjson
          tags => ['ucidc']
        }
}
input {
        tcp {
          port => 2516
          codec => json_lines
          type => nxlogjson
          tags => ['ucidc']
        }
}
input {
        tcp {
          port => 2517
          codec => json_lines
```

```
                type => nxlogjson
                tags => [’ucidc’]
              }
        }
        filter {
        mutate {
        remove_field => [
          "nxlog_input",
          "SourceModuleName",
          "type",
          SourceModuleType
         ]}
        if [EventID] == 4624 {
            mutate {
              remove_field => ["Message"]
            }
        }
        }
        #output {
        #    stdout { codec => rubydebug }
        #  }
        output {
           if "ucidc" in [tags] {
              elasticsearch {
                index => "windowsad-%{+YYYY.MM.dd}"
                host => "1.2.3.4"
                #template=> "/opt/logstash/wtemplate.json"
                manage_template=> false
                #protocol => ’http’
              }
           }
}
```

Anexo 12. Interfaz de la aplicación web de monitoreo Kibana

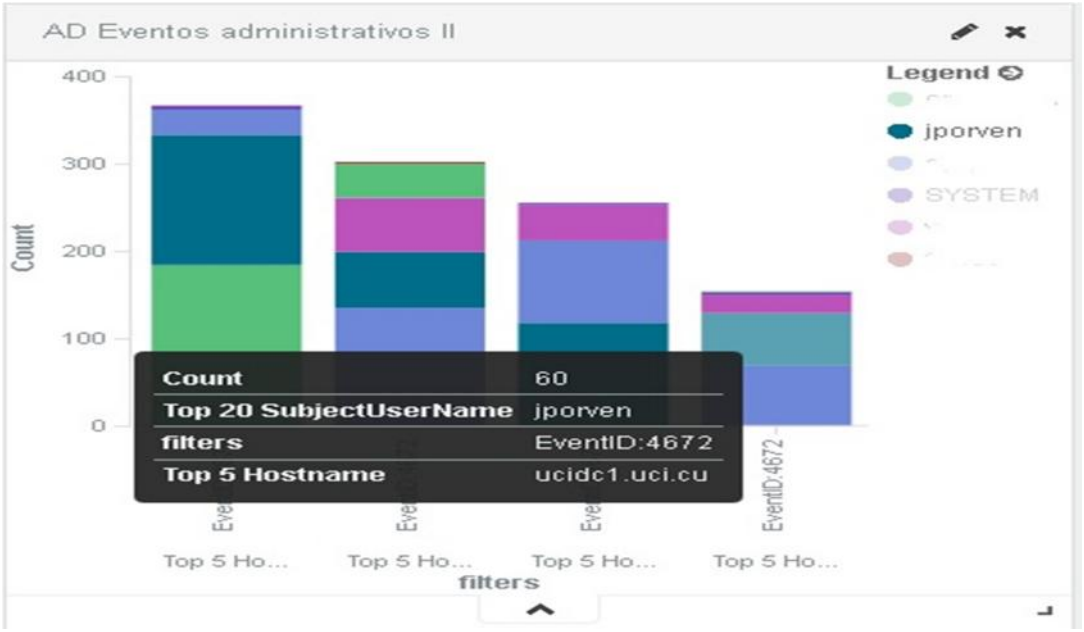
Interfaz de configuración de Kibana donde se añaden los índices a procesar, se crean los paneles interactivos y se agrupan en paneles de mando.



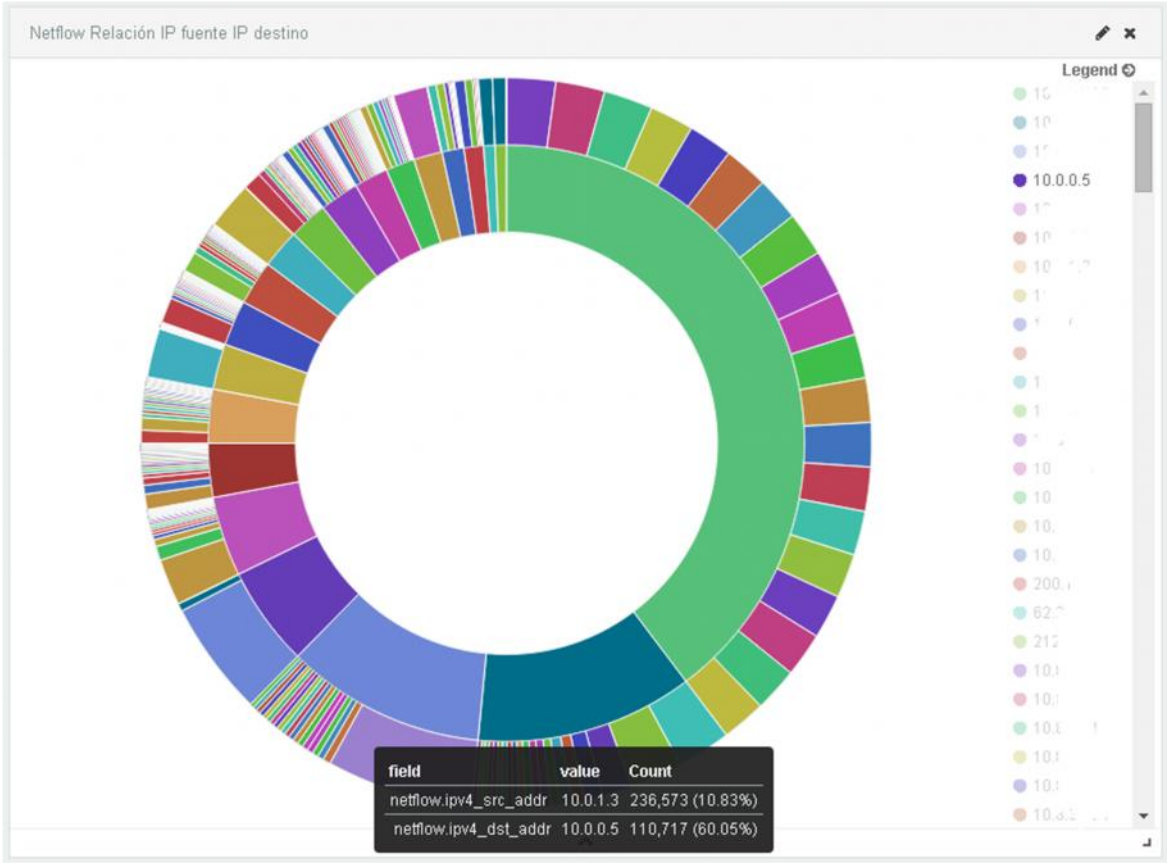
Anexo 13. Gráficas que componen los paneles de mando generados en Kibana donde se agrupan los principales registros de eventos obtenidos de las trazas

Tipos de eventos registrados por *auditd* en servidores Linux en un período de tiempo determinado, agrupados por cantidad, representados en una gráfica de barras.

Registro de eventos generados por acciones con privilegios administrativos en los servidores de autenticación.



Relación de tráfico Netflow por IP fuente y IP destino.



Ejemplo de agrupación de tráfico localizado por IP y región destino filtrado para una región determinada.