Universidad de Cienfuegos “Carlos Rafael Rodríguez”

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA INFORMÁTICA

Sistema de Gestión de Logs para la División Territorial de ETECSA en Cienfuegos

---

Autor: Lázaro Andrés Rodríguez Ramírez

Tutores: Msc. Karina Leonor Fernández Sánchez

Msc. Ciro Rodríguez León

Pensamiento:

Para obtener éxito en el mundo, hay que parecer loco y ser sabio.

- Barón de Montesquieu-

Dedicatoria:

A mi familia y especialmente a mi mamá, a mi papá , a mis abuelos y a mi tío que me facilitó los medios para este duro trayecto.

Agradecimientos:

A mis padres, sin ellos no hubiese sido posible llegar hasta aquí. A los profesores que contribuyeron en todos estos años a mi formación profesional. A mis tutores Msc. Karina Leonor Fernández Sánchez Msc. Ciro Rodríguez León que ambos tuvieron la paciencia y dedicación necesaria para que este trabajo fuera posible. A Omar Padrón que fue una persona de gran ayuda en este proyecto. A mis compañeros de grupo por todo su apoyo y ayuda en estos años compartidos. A cada persona que de una forma u otra contribuyó con su esfuerzo y ayuda al éxito de este trabajo de diploma.

Resumen:

Sistema de Gestión de Logs para la División Territorial de ETECSA en Cienfuegos.
Uno de los objetivos de este proyecto es señalar la importancia de la informacón que aportan los ficheros logs de los diferentes programas, aplicaciones y equipos, y de cómo su conocimiento y análisis ayauda a la gesión de la seguridad de la Organización.

Se presenta un diseño de red, cuyo objetivo es facilitar la recopilación y filtrado de los ficheros logs, para esto se apoya de diferentes herramientas de código abierto, como Grafana, Elasticsearch, Graylog, InfluxDB y clientes para la obtención de los logs de los diferentes equipos y aplicaciones. Con estas herramientas se puede monitorear los diferentes eventos que realizan los equipos y las aplicaciones.

Palabras clave: logs, diseño, monitorear, gestión.

Abstract:

Log Management System for the Territorial Division of ETECSA in Cienfuegos. One of the objectives of this project is to point out the importance of the information provided by the log files of the different programs, applications and equipment, and how their knowledge and analysis aids the management of the Organization's security.
A network design is presented, whose objective is to facilitate the collection and filtering of the logs, for this is supported by different open source tools, such as Grafana, Elasticsearch, Graylog, InfluxDB and clients to obtain the logs of the different equipment and applications. With these tools we can monitor the different events that teams and applications perform.
Keywords: logs, design, monitor, management.

Índice de Contenido

Introducción:

Uno de los aspectos de gran importancia relacionado con la seguridad de la información, es entender que esta debe ser gestionada, sobretodo cuando se trata de la protección de las redes corporativas [1]. En la actualidad las redes empresariales pueden contener amplia variedad de sistemas y equipos capaces de generar logs de gran interés.

Las mejoras tecnológicas en los últimos años han sido de gran impacto en el ámbito personal como profesional. Hoy en día las empresas se apoyan en los sistemas que generan una gran cantidad de datos en forma de trazas textuales, llamadas técnicamente “Logs”. Esta información es oculta para el usuario pero suele estar relacionada con su actividad informática (por ejemplo, historial de navegación, programas abiertos, etc.) o con los propios sistemas de información (es decir, seguridad, estado actual de programas, accesos, conectividad de redes, etc.), en efecto, cada vez más estas soluciones se están convirtiendo en herramientas principales para la toma de decisiones e identificar un problemas de negocio [2].

Los Logs, nos permiten comprender el comportamiento de los sistemas o programas, estos suelen escribirse en ficheros. Aunque se pueda examinarlos más adelante, estos archivos por sí solos no nos permitirán detectar si se ha producido un error [3].

La aplicación de buenas prácticas en la gestión de los Logs aporta distintos beneficios, tanto a nivel de funcionamiento de los sistemas como de objetivos de negocio. Entre otras ventajas, te permitirá:

- Aplicar técnicas de Machine Learning para detectar patrones que te ayuden a tomar decisiones de negocio.
- Mejorar la gestión y el control de la información facilitando así el acceso y la explotación de estos datos.
- Detectar amenazas de red o virus para poder actuar en consecuencia con rapidez.
- Prevenir fugas de información, así como comportamientos inadecuados que causen errores.

Introducción

En un escenario habitual, se suelen ejecutar multitud de aplicaciones al mismo tiempo. Su trazabilidad se reduce, en el mejor de los casos, a Logs persistidos en el disco duro de la máquina y, en el peor, a simples trazas de texto por pantalla tipo “print” que desaparecen a los pocos segundos [4].

Si no se tiene un almacenamiento adecuadado de los Logs, pueden producirse diversos problemas en la actividad del negocio:

- Visibilidad nula de errores para los equipos de desarrollo y/o sistemas. O sea, será difícil detectar problemas y solucionarlos con rapidez.
- Metodología de trabajo no estandarizada, es decir, cada usuario aplicará su propia forma basada en su experiencia.
- Accesos e información descentralizada que causará dificultades para trabajar con estos datos.
- Aumento del tiempo de respuesta ante una incidencia que afectará a niveles de servicio o SLAs (en relación con el nivel de calidad).

Existen distintas herramientas libres (Open Source) para obtener una buena gestión de Logs, que nos pueden ayudar a almacenarlos, así como sistemas comerciales que nos permitirán gestionar y visualizar los datos. Con la combinación de ambos elementos, se puede construir sistemas que centralicen y exploten la información para identificar conflictos y tomar mejores decisiones de negocio [5].

Para garantizar una adecuada protección de Logs, será necesario almacenarlos de forma sesgura en un entorno distinto al sistema protegido, para evitar que los intrusos puedan modificar o eliminar: grabación de los registros en discos WORM (Write Once Read More), generación de una copia en papel, etcétera.

En algunos casos se puede recurrir a una gestión centralizada de Logs, mediante un servidor que se encargue de guardar copia de todos todos los registros enviados por los dispositivos de red y los servidores. Para ello, se podrían utilizar aplicaciones como syslog[1] para

1 Syslog: Es un estándar pararegistro de mensajes

centralizar los registros. De este modo, se refuerza la seguridad frente a intrusos que pretendan eliminar su rastro manipulando los Logs de los equipos.

Además, un servidor centralizado de Logs permite coservar los registros durante un mayor período de tiempo, lo que facilita el análisis detellado de estos registros, incluyendo el estudio de la relación entre eventos incluidos en los Logs de distintos equipos y dispositivos (elementos de red, herramientas de seguridad).

Es conveniente mantener todos los relojes de los equipos y dispositivos perfecamente sincronizados para poder comparar los distintos tipos de Logs. Se puede utilizar el protocolo NTP (Network Time Protocol, www.ntp.org), el cual nos permite una perfecta sincronización.

Los administradores de red tienen a su disposición una serie de herramientas para analizar toda la información registrada en los Logs, en las que se encuentran distintos tipos de filtros y aplicaciones que permiten detectar de forma automática patrones de ataques o situaciones de potencial peligro.

En este sentido, se debería considerar el problema de exceso de información registrada en el sistema pueda llegar a desbordar a sus administradores, provocando una situación bastante habitual en muchas organizaciones: que los datos de los registros de actividad no sean analizados de forma adecuada.

Por otra parte, suele ser muy recomendable realizar un estudio previo del tráfico en la red para facilitar la posterior detección de situaciones de intrucción.

Un servidor Syslog proporciona una herramienta rica en características para ayudarlo a administrar sus necesidades de registro del sistema, incluidas las capacidades de exportación mejoradas. Vea los mensajes en tiempo real o filtre los datos de resultados de la manera que necesita verlos. Tome las riendas de su red al comprender los datos que le proporcionan sus dispositivos.

La auditoría de red genera volúmenes de datos de registro de varios sistemas y dispositivos. Además de los mensajes syslog generados por los sistemas operativos y las aplicaciones, los mensajes también se generan mediante procesos en impresoras, enrutadores, concentradores, conmutadores y otros dispositivos en red. Es una buena práctica enviar

todos estos mensajes a un servidor syslog central para su almacenamiento y revisión. Un servidor syslog debe ofrecer funciones que lo ayuden a monitorear y administrar de manera eficiente los mensajes syslog para correlacionar eventos en su red. Esto permite tener la relativa seguridad de que la información que se está manejando en la gestión de incidente es veraz. Además, tener los registros duplicados en el sistema de origen y en el servidor *syslog* permite *a posteriori* comparar ambos, y detectar así posibles modificaciones que proporcionen información sobre el atacante [6].

Situación Problémica:

El desarrollo de las telecomunicaciones está cada vez más relacionado a la Informática. En las redes TCP/IP conviven computadoras, sistemas, equipos de telecomunicaciones, etc que generan logs y necesitan ser gestionados. En la División Territorial de ETECSA en Cienfuegos (DTCF) existe una gran diversidad tecnológica, pero los logs generados no son analizados de forma centralizada. Dicho de otro modo, no cuenta con un sistema de gestión de registros de eventos a pesar su importancia para para obtener un mejor control de los logs.

ETECSA está inmersa en una migración a software libre por lo que es un prerequisito que la solución se base en estas tecnologías. En la DTCF la solución manejará grandes volúmenes de datos, teniendo que analizar sintácticamente los logs para su posterior análisis.

Por tanto, se plantea como **problema científico:** ¿cómo implementar una gestión de registros de eventos (logs) en la División Territorial de ETECSA en Cinfuegos?

Como **objeto de estudio** la gestión de logs con herramientas de software libre y como **campo de acción** los logs de diferentes sistemas, servicios y equipos en la red la División Territorial de ETECSA en Cienfuegos.

El **Objetivo general** de esta investigacion: diseñar e implementar un sistema de gestión de logs para la División Territorial de ETECSA en Cienfuegos utilizando herramientas de software libre.

Objetivos específicos

1. Identificar tipos de logs generados en una red TCP/IP y los sistemas que existen para su gestión.

2. Caracterizar la red de la División Territorial de ETECSA en Cienfuegos en cuanto a logs generados por sistemas, servicios y equipos.

3. Diseñar un Sistema de Gestión de logs en la División Territorial de ETECSA en Cienfuegos.

4. Implementar la solución propuesta.

5. Evaluar el desempeño del sistema implementado.

Se plantea como **idea a defender**: la utilización de un Sistema de Gestión de logs basado en herramientas de software libre, pudiera ser un modo efectivo para poder controlar los registros de eventos en la red de la División Territorial de ETECSA en Cienfuegos.

Tarea de investigación

- Revisión de los conceptos necesarios y las herramientas para el desarrollo del Sistema de Gestión de logs.
- Identificación de dispositivos y aplicaciones que generen logs en la División territorial de ETECSA en Cienfuegos.
- Diseño del Sistema de Gestion de logs a implementar en la División territorial de ETECSA en Cienfuegos.
- Implementación del prototipo de Sistema de Gestión de logs en la División Territorial de ETECSA en Cienfuegos.
- Análisis de los resultados obtenidos de la solución propuesta.

El **aporte práctico** está dado porque, con la implementación del Sistema de Gestión de logs, se tendrán argumentos para la toma de decisiones y detectar oportunamente eventos de seguridad informática en la División Territorial de ETECSA en Cienfuegos.

La tesis esta estructurada, en Resumen, Introducción, 3 capítulos, Conclusiones,

Recomendaciones, Bibliografía y Anexos.

Resumen capitular.

Capítulo 1: Sistemas de Gestión de Logs.

Se describen las características de los diferentes tipos de logs tanto de sistemas operativos, servicios de red o equipos que pertenecen a una red TCP/IP. Analizando las herramientas más efectivas para su gestión que son libres.

Capítulo 2: Diseño de un Sistema de Gestión de logs para la División Territorial de ETECSA en Cienfuegos.

Se determinan las herramientas para la implementación del Sistema de Gestión de logs deseado y se describe la solución propuesta para la División Territorial de ETECSA de Cienfuegos.

Capítulo 3: Evaluación del Sistema de Gestión de logs mediante la simulación de eventos y análisis de los resultados.

Se realiza una evaluación del sistema propuesto a partir de crear los escenarios necesarios que generaran logs que es interés de la Entidad no pasen inadvertidos.

Capítulo 1 Sistemas de Gestión de Logs.

Introducción.

No existe una herramienta que implementada por si sola gestione de manera efectiva todos los eventos posibles a monitorear en una red. Se puede encontrar gran diversidad de ellas tanto open source o propietarias, pero la selección depende de qué se quiere lograr [7][8].

En la División Territorial de ETECSA en Cienfuegos (DTCF) se desean monitorear: métricas de servidores, logs del squid proxy y eventos locales asociados a la autenticación de usuarios pero haciendo uso de herramientas open source. Para satisfacer estos requisitos fueron seleccionadas un conjunto de aplicaciones que serán analizadas a continuación, así como otros conceptos necesarios para la comprensión de los resultados de este trabajo.

1.1 Logs de sistemas operativos.

Las empresas se apoyan en sistemas que generan una gran cantidad de datos en forma de ficheros, llamadas técnicamente Logs. Esta información no es visible para el usuario pero suele estar relacionada con su actividad informática, éstos nos explican el comportamiento de los sistemas o programas. Aunque más adelante pueden ser examinados, estos archivos por si solos no nos permitirán detectar si se ha producido un error [9].

1.1.1 Logs en Windows.

Cuando se busca información sobre registros en Windows, lo primero que se muestra son los registros de eventos y Visor de eventos [10].

Los registros de eventos son archivos especiales que registran los eventos importantes que tienen lugar en el equipo, ejemplo, cuando un usuario inicia una sesión en el equipo o cuando se produce un error en un programa. Siempre se produce este tipo de evento, el sistema operativo Windows los va incluyendo en un registro de eventos que se puede leer mediante el Visor de eventos [11].

La herramienta Visor de eventos, nos mostrará los registros de los eventos transcurridos en el equipo. Los registros de Windows incluyen.

- **Eventos de aplicaciones.** Muestran los errores, advertencias o información, dependiendo de su gravedad, que se derivan del uso de las aplicaciones instaladas en nuestro equipo. Un error es un problema importante, como una pérdida de datos. Una advertencia es un evento que no es importante necesariamente, pero puede indicar la posibilidad de problemas en el futuro. Un evento de información describe la operación correcta de una aplicación, un servicio o un controlador.
- **Eventos de configuración.** Muestran los registros relacionados a los equipos que se han configurado como controladores de dominio.
- **Eventos de seguridad.** Son eventos conocidos como auditorías y se describen como correctos o con error dependiendo del evento, ejemplo, si un usuario logra iniciar una sesión en Windows correctamente o no.
- **Eventos del sistema.** muestran los registros relacionados al funcionamiento del sistema operativo y sus servicios, estos pueden clasificarse al igual que los eventos de aplicaciones, en errores, advertencias o informaciones.
- **Eventos reenviados.** Los eventos se reenvían a este registro desde otros equipos.

Conociendo los distintos tipos de eventos se puede detectar con mayor facilidad el tipo de problema que se tiene y en base a ello buscar una solución acorde. Los registros de servicios y aplicaciones pueden variar. Estos incluyen registros independientes para los programas que se ejecutan en el equipo, así como registros más detallados relacionados con servicios específicos de Windows.

1.1.2 Logs en Linux – Unix.

Los archivos log registran las actividades de los programas y servicios en sistemas GNU/Linux. Ellos son herramientas fundamentales a la hora de administrar y/o analizar lo que ocurre en el sistema operativo. Es un mecanismo estándar que se encarga de recoger los mensajes generados por los programas, aplicaciones y demonios y enviarlos a un destino predefinido. En cada mensaje consta la fuente (el programa que generó el mensaje), la prioridad (nivel de importancia del mensaje), la fecha y la hora [12].

Hay varios niveles de prioridad de los mensajes (de menos a más prioritario: debug, info, notice, warning, warn, err, error, crit, alert, emerg y panic) y varios tipos de mensajes (auth, authpriv, cron, daemon, kern, lpr, mail, mark, news, security, syslog, user, uucp y local0-local7). [13]

Archivos de registro comunes (pueden variar según la distribución):

La carpeta «/var/log/», contiene toda la lista de archivos logs del sistema operativo.

Se puede ver que se escribe en cada uno de estos archivos leyendo el fichero de configuración «/etc/rsyslog.conf». Aunque muchos programas manejan sus propios logs y los guardan en /var/log/<programa>. Además, es posible especificar múltiples destinos para un mismo mensaje.

Conclusión, en /var/log se almacenan todos los registros del sistema. No obstante, algunas aplicaciones como httpd incluyen ahí dentro un subdirectorio en el que almacenan sus propios archivos de registro.

- syslog es quizá el fichero log más importante del sistema, localizado en /var/adm ó /var/log. Se guardan mensajes relativos a la seguridad de la máquina,
- /var/log/messages ó /var/adm/messages registro de mensajes generales del sistema. Contiene mensajes del sistema en general, incluyendo los del inicio del mismo. Incluye trazas de mail, cron, kern, auth, y muchísimos más. Es uno de los más importantes y que más información tienen. Se Encuentra los logs que llegan con prioridad info (información), notice (notificación) o warn (aviso).
- /var/log/auth.log: log de autenticación. En este log se registran los login en el sistema. Los intentos fallidos se registran en líneas con información de tipo invalid pasword o authentication failure. Conexiones al sistema sistema incluidos los intentos fallidos y los accesos como root. Información sobre eventos de autentificacion de usuarios.
- /var/adm/sulog: se registran las ejecuciones de su, indicando fecha, hora, usuario que lanza el programa y usuario cuya identidad adopta, terminal asociada indicando si ha sido realizado con éxito o no.

- /var/log/kern.log: registro del kernel. Información que proporciona el kernel. Ayuda para arreglar problemas relacionados.
- var/log/cron.log: registro de cron.
- var/log/maillog: registro del servidor de correo.
- var/log/qmail: registro de Qmail.
- var/log/httpd: registro de errores y acceso Apache.
- var/log/boot.log: registro de inicio del sistema.
- var/log/mysqld: registro de basa de datos MySQL.
- var/log/secure: log de autenticación. Información relacionada con autenticación y privilegios incluidos del SSH.
- var/log/utmp or var/log/wtmp: registro de logins. O utilaza el comando “who” para indicarnos quien esta conectado, información relativa a cada conexión y desconexión al sistema, con información de usuarios conectados.
- var/adm/loginlog: en algunas de las versiones Unix (como Solaris), si se crea el archivo login, cuando se produzcan cinco o mas intentos seguidos se registrarán en él.
- var/log/dmesg: en este archivo se almacena la información que genera el kernel durante el arranque del sistema.
- var/log/daemon.log: información de varios daemons en segundo plano que funcionan en el sistema.
- var/log/Xorg.0.log: mensajes de las X. Información sobre el entorno gráfico.
- var/log/rpmpks: contiene información cuando un paquete RPM es intalado o removido.
- var/log/cups: todos los mensajes relacionados con el sistema de impresión.
- var/log/lastlog: muestra para cada usuario cada cuando fue la última vez que entro en el sistema.
- var/log/btmp: intentos de ingresos fallidos.

- var/log/anaconda.log: todos los mensajes relacionados con la instalación del sistema.
- var/log/faillog: contiene intentos fallidos de login, se usa con el comando faillog para ver los resultados.
- var/log/yum.log: información de los paquetes instalados con yum.
- var/log/debug: información de depuración de los programas.
- var/log/fontconfig.log: configuración de las fuentes del sistema.
- var/log/samba: en este directorio se guarda los logs relacionados con el servicio samba, el cual permite la conexión de Windows con Linux.

Otros logs:

- El archivo “/etc/sudoers”, indica indica cuál usuario puede ejecutar comandos como administrador (mediante los comandos “su” o “sudo”).
- El archivo “.bash_history”, el cual se se encuentra en la carpeta de usuario y almacena el historial de comandos ejecutados en la consola del equipo.
- La carpeta “/var/spool/cron/crontabs” en la cual se almacenan las tareas programadas de cada uno de los usuarios del sistema.
- /var/adm/debug: información de depuración (de debug) de los programas que se ejecutan en la máquina, este se puede enviar por el sistema operativo opor la aplicaciones.
- X11- Sistema X Windows: /usr/binX11/xauth: recoge errores de autenticación.
- X11- Los logs de Xorg se localizan en /var/log/Xorg.x.log: información de cada dispositivo al que el servidor X11 se conecta.

1.1.3 Otros logs.

No solo los sistemas operativos generan logs, algunos ejemplos ilustrativos pueden ser [14]:

- Servidor web: un log de acceso puede ser útil para identificar el número de visitantes, los dominios desde los que están visitando, el número de peticiones para cada página,

los patrones de uso según el día de la semana o incluso la hora del día.

- Microsoft Exchange: los logs de transacciones son archivos que se utilizan para transmitir información (mensajes de correo electrónico, nuevos usuarios, carpetas borradas, etc.) a la base de datos de Exchange. Todo se envía primero al log de transacciones y luego a la base de datos cuando el sistema lo permite.
- Los routers de red: los archivos de log registran los procesos que fallan, las conexiones y desconexiones de los servicios y dispositivos WAN, el estado de las conexiones VPN, etc.
- Firewall: los archivos de log registran qué conexiones de red se permitieron y se eliminaron.

1.2 Sistemas de Gestión de Logs.

Se entiende por Sistema de gestión de Logs, aquel que es capaz de almacenar e interpretar logs. El alcance que tenga este sistema depende de qué se desea supervisar. Puede enfocarse a una aplicación en especial o contemplar gran variedad de eventos de una red.

Hoy en día las herramientas de monitorización más exitosas tienen entre sus opciones la posibilidad de monitorizar logs. Además se puede encontrar en el mercado muchas herramientas analíticas y de gestión de logs.

Estos sistemas ayudan a:

- Monitorización de aplicaciones, redes y servidores
- Optimización y errores de depuración
- Análisis forense y análisis de causa raíz.
- Evaluación de vulnerabilidad.
- Cumplimiento de la normativa legal o log de eventos con fines de auditoría.
- Definir planes de capacidad o cambios arquitectónicos.

1.2.1 Principales aspectos que se han de tener en cuenta cuando se desida empezar

un proyecto de monitorizacion de logs.

Define un objetivo para el análisis de archivos de log.

Dado que el análisis de archivos de log puede contribuir con diferentes actividades, un punto crucial es definir la meta y los objetivos que se quiere alcanzar con este proyecto. Tal vez nuestro interés se centra en mejorar nuestras capacidades de resolución de problemas y reducir el tiempo necesario para determinar la causa fundamental de los problemas que puedan surgir.

La seguridad puede ser nuestro principal interés, o cubrir los requisitos impuestos por algunas organizaciones reguladoras.

Cuando se inicia un proyecto de monitorizacion de log, tener una meta clara nos permite dimensionar correctamente el proyecto y tomar decisiones correctas durante la evaluación de las herramientas.

Luego hay que definir los objetivos en estos temas específicos:

- **Visibilidad:** Especifique si está interesado en tener un panel de control que le permita acceder a la información de log o si está más interesado en informes personalizados.
- **Accesibilidad**: ¿Cuán importante para nuestro objetivo es el análisis cruzado? ¿Cuán vital es la posibilidad de realizar diferentes consultas sobre los datos de los logs?
- **Integración**: ¿Se tiene ya una herramienta de monitorización o una herramienta de gestión de logs? ¿Cómo se quiere integrar estas herramientas en nuestro proyecto?
- **Alertas**: ¿Es crucial la gestión de alertas cuando se detectan ciertos eventos o patrones?
- **Escalabilidad**: ¿Cómo se cree que los requisitos de monitorizacion de logs crecerán a corto y mediano plazo? ¿En dispositivos o sistemas? ¿Con diferentes objetivos?

Lista de archivos de log.

Es importante definir una primera lista de archivos de log que deben ser recopilados y analizados, incluyendo la siguiente información sobre cada log:

- **Tipo**: Sistema Operativo, herramientas de identificación, red, Aplicaciones, seguridad de endpoints, etc.
- **Sistemas**: Servidor Windows 2008 o servidor Linux, Active Directory, LDAP, Servidor DHCP, Router Cisco 1941, Cisco ASA, VPN, seguridad de puntos finales MacAfee, servidor Apache, Servidor JBoss, etc.
- **Identificación del log:** nombre y ubicación.
- **Tamaño**: tamaño de log en MB.
- **Descripción del log:** breve descripción de los eventos registrados en este archivo de log.

Por ejemplo:

Tipo	OS
Sistema	Windows Server 2008 R2
Nombre y dirección Ip	MDRServer / 191.168.4.56
Log	Setup / Systemroot\System32\Winevt\logs\setup.evtx
Tamaño	1028 MB
Descripción	Registra eventos durante la configuración y estadísticas de rendimiento

Evaluar las herramientas.

La recomendación es evaluar las características de las herramientas de monitorización con los objetivos y necesidades existentes.

En este punto se debe considerar que se puede encontrar herramientas con diferentes enfoques para la monitorizacion de logs:

- Software como servicio (SaaS) que no requiere instalaciones sino la contratación de un servicio remoto.
- Herramientas instaladas localmente.
- Herramientas basadas en dispositivos de hardware.

1.3 Herramientas para la gestión de Logs.

Una herramienta de monitoreo de redes resulta esencial en el entorno corporativo con el objetivo de asegurar el funcionamiento de los sistemas informáticos, así como para evitar cualquier tipo de fallos en la red. Además, este tipo de software resulta de gran ayuda en la optimización de la red ya que aporta un amplio abanico de información sobre el uso de los diferentes recursos de la misma. [15][16][17]

1.3.1 Clientes.

Existen diferentes programas encargados de recolectar logs y datos de la computadora o sistema para eviarlos a los colectores externos que forman parte del Sistema de Gestión de Logs.

A continuación serán analizados el filebeat y winlogbeat usados para colectar logs en sistemas linux y windows respectivamente. Además del telegraf para recolección de métricas [18].

1.3.1.1 Filebeat.

Filebeat es un cargador de datos de registro para archivos locales. Instalado como un agente en sus servidores, supervisa los directorios de registro o archivos de registro específicos, los agrupa y reenvía a Elasticsearch o Logstash para la indexación [19][20].

Filebeat consta de dos componentes principales: buscadores y cosechadores . Estos componentes trabajan juntos para rastrear archivos y enviar datos de eventos a la salida que especifique.

Una cosechadora es responsable de leer el contenido de un solo archivo línea por línea, y envía el contenido a la salida. Se inicia una para cada archivo.

Un prospector es responsable de administrar a los recolectores y encontrar todas las fuentes para leer. Si el tipo de entrada es el log, el prospector encuentra todos los archivos en la unidad que coinciden con las rutas globales definidas e inicia una cosechadora para cada archivo. Cada prospector se ejecuta en su propia rutina

Filebeat garantiza que los eventos se entregarán a la salida configurada al menos una vez y sin pérdida de datos. Puede lograr este comportamiento porque almacena el estado de entrega de cada evento en el archivo de registro. En situaciones donde la salida definida está bloqueada y no ha confirmado todos los eventos, Filebeat seguirá tratando de enviar eventos hasta que la salida reconozca que los ha recibido.

1.3.1.2 Winlogbeat.

Winlogbeat envía registros de eventos de Windows a Elasticsearch o Logstash. Puede instalarlo como un servicio de Windows en Windows XP o posterior [21].

Winlogbeat lee desde uno o más registros de eventos mediante las API de Windows, filtra los eventos según los criterios configurados por el usuario y luego envía los datos del evento a las salidas configuradas (Elasticsearch o Logstash). Mira los registros de eventos para que los nuevos datos del evento se envíen de manera oportuna. La posición de lectura para cada registro de eventos se conserva en el disco para permitir que Winlogbeat se reanude después de reiniciar.

Puede capturar datos de eventos de cualquier registro de eventos que se ejecute en su sistema. Por ejemplo:

- Aplicación
- Hardware
- Seguridad
- Sistema

1.3.1.3 Telegraf.

Telegraf es un agente para recolectar y reportar métricas y datos. Es parte de la pila TICK y es un agente de servidor impulsado por complementos para recopilar e informar métricas. Telegraf tiene complementos o integraciones para obtener una variedad de métricas directamente del sistema en el que se ejecuta, extraer métricas de API de terceros o incluso escuchar métricas a través de los servicios para consumidores de StatsD y Kafka. También tiene complementos de salida para enviar métricas a una variedad de otros almacenes de datos, servicios y colas de mensajes, incluidos InfluxDB, Graphite, OpenTSDB, Datadog, Librato, Kafka, MQTT, NSQ y muchos otros [22].

Puede ser utilizado para:

- Bases de datos: Conectarse a fuentes de datos como MongoDB, MySQL, Redis para recopilar y enviar métricas.
- Sistemas operativos: Recopila métricas del sistema como CPU, memoria, E / S y más.
- Sensores de IoT: Recopila datos de sensores de diferentes dispositivos de IoT.

1.3.2 Indexar eventos.

Los eventos generados por equipos y aplicaciones se caracterizan por ser de gran volumen y variedad. La acción de clasificarlos y almacenarlos indexados optimiza el proceso de análisis aunque puede constituir un trabajo complejo.

1.3.2.1 Logstash.

Logstash centraliza, transforma y almacena tus datos, es una fuente de procesamiento de datos de código abierto. Del lado del servidor puede adquirir datos de una multitud de fuentes simultáneamente, las transforma y luego las envía a Elasticsearch [23].

A medida que los datos viajan de la fuente al servidor, los filtros de Logstash analizan cada evento, identifican los campos para construir la estructura y los transforman para converger en un formato común que permita un análisis rápido y más fácil.

Logstash permite:

- Derivar estructura de datos no estructurados con grok.
- Descifrar las coordenadas geográficas de las direcciones IP.
- Facilitar el procesamiento general independientemente de la fuente de datos, el formato o el esquema.
- Enrutar los datos donde lo desee, no tiene obligatoriamente que a Elasticsearch.

1.3.2.2 Graylog.

Graylog proporciona un sistema unificado y centralizado de mensajes procedentes de diferentes fuentes: sistema operativo, servidores de aplicación, sistemas de información, etc. Dispone de un sistema de alertas y de búsqueda de histórico de logs usando ElasticSearch como base de datos de índices [24].

Cuenta con una interfaz web amigable que facilita el trabajo de crear los índices y analizar el comportamiento de los logs recibidos. Se puede implementar en un solo servidor o en un cluster para evitar la pérdida de disponibilidad.

Permite, además, realizar consultas sobre los datos, crear cuadros de mando, alarmas y muchas otras funcionalidades interesantes en entornos de TI y programación [25]. Se compone de tres componentes fundamentales:

- Mongo DB: Es el almacén de configuraciones y metadatos.
- ElasticSearch: Actúa como motor de búsquedas y almacenamiento.
- Graylog Server: Es el propio servidor de Graylog que incluye todas sus funcionalidades y una interfaz de usuario web (Graylog UI).

1.3.3 Almacenar logs.

La selección de dónde almacenar los logs de manera que las búsquedas sean efectivas es un aspecto fundamental en el diseño de un Sistema de Gestión de Logs. Es Elasticsearch una opción excelente como se aprecia a continuación.

1.3.3.1 Elasticsearch.

Elasticsearch es un motor de búsqueda en tiempo real y de analítica. Permite explorar datos a una velocidad y a una escala que antes no era posible. Se utiliza para búsqueda de texto completo, búsquedas estructuradas, analítica en tiempo real, o una combinación de las tres [26].

Principales características.

- Datos en tiempo real: Los datos en una aplicación están en constante cambio. Elasticsearch permite disponer en tiempo real de los últimos cambios realizados sobre los datos.
- Distribuido: Elasticsearch está preparado para escalar horizontalmente. Con Elasticsearch se puede empezar con una solución pequeña e ir creciendo a medida que crece el negocio. En la medida en que se necesite más capacidad simplemente habrá que añadir más nodos y dejar que el cluster se reorganice para aprovecharse por sí solo del hardware extra.
- Alta disponibilidad: Los clusters de Elasticsearch son flexibles. Son capaces de detectar y eliminar nodos que estén fallando y reorganizarse a sí mismos para asegurar que los datos estén a salvo y permanezcan accesibles.
- Multi-tenancy (Multitenencia): Un cluster de Elasticsearch puede alojar múltiples índices, que podrán ser consultados de manera independiente o en grupo. Además, asignando alias a los índices, permite añadir índices al vuelo de manera transparente para la aplicación.
- Búsquedas full-text: Elasticsearch se basa en Lucene para proveer con las más poderosas capacidades de búsqueda de texto. Las búsquedas soportan multidioma, geolocalización, autocompletado, etc.
- Orientado a documentos: Las entidades se almacenan en Elasticsearch como documentos JSON estructurados. Todos los campos son indexados por defecto y todos los índices pueden ser usados en una misma consulta.

- Gestión de conflictos: Elasticsearch provee mecanismos (optimistic version control) para asegurar que los datos nunca se pierdan debido a cambios simultáneos sobre un mismo documento realizados por diferentes procesos.
- Sin esquemas: Elasticsearch permite trabajar sin esquemas que definan la estructura de los datos. Simplemente pasándole un documento JSON intentará detectar la estructura de los datos, indexarlos y hacer que sean accesibles para las búsquedas.
- API Restful: Elasticsearch proporciona un API Restful, con JSON y sobre HTTP, que permite realizar casi cualquier acción. También existen APIs para otros lenguajes, entre los que se encuentra java.
- Persistencia a nivel de operación: Elasticsearch vela por la seguridad de los datos. Todos los cambios que se realizan sobre los documentos se almacenan en logs de transacciones en múltiples nodos del cluster para minimizar las pérdidas de información.
- Open source: Es open source y está bajo la licencia Apache 2. Permite descargar, usar y modificar el código.

Para administrar elasticsearch se puede usar *Cerebro,* una herramienta de administración web para elasticsearch, de código abierto.

1.3.3.2 InfluxDB.

Influxdb es un servidor de base de datos de series de tiempo (timeseries), ideal para logs o datos para gráficas que se generen en vivo. Programado en go permite la interacción via API HTTP(S) (JSON) e interficie web y los datos de gestionan con un lenguaje similar a SQL [27].

En las tablas no hay que tener una previsión de las columnas que tendrá, se pueden añadir con facilidad. Con un simple comando curl via CLI se puede realizar la inserción de datos [28].

Algunas de las características que definen InfluxDB son las siguientes:

- Se asume que si se envía el mismo dato varias veces, es el mismo dato por lo que se aplica la política de resolución de conflictos (de forma resumida, si son exactamente

los mismos datos de tags set, field set, timestamp, se sobreesciben los valores en field set con los datos del último Point) por lo que en ciertos casos, se pueden perder datos.

- El borrado de datos es una situación extraña. Normalmente se borran datos antiguos.
- Se limita la funcionalidad de borrado para incrementar las de escritura y lectura.
- La actualización de datos también es una situación poco común, por lo que su funcionalidad está restringida.
- La mayoría de los datos tienen timestamps recientes y se guardan en orden ascendente para mejorar el rendimiento.
- La base de datos puede gestionar un gran volumen de lecturas y escrituras, priorizandolas sobre la vista de los datos.
- No está soportado el uso de joins entre tablas

1.3.4 Graficar análisis.

Una vez que se ha logrado enviar los logs adecuados y éstos han sido almacenados con los índices correctos, solo resta mostrar los resultados de manera eficiente.

Existen diversos programas que permiten graficar a partir de entradas Elasticsearch e InfluxDB. En el proceso de selección no solo debe tenerse en cuenta la estética sino la posibilidad de que en una sola se puedan mostrar resultados de diferentes fuentes.

1.3.4.1 Kibana.

Kibana es un sistema analítico open source (licencia Apache) que permite hacer análisis y búsquedas sobre Logstash y otros sets de datos timestamped almacenados en ElasticSearch. Permite comenzar a analizar los logs almacenados en ElasticSearch de inmediato [29].

Ofrece un interface muy potente sobre Logstash pero también permite crear dashboards a medida, con características como personalización, selección de rangos, drill down, además de poder compartirse y guardarse.

1.3.4.2 Grafana.

Grafana es una tecnología de código abierto que le permite consultar, visualizar, alertar y comprender sus métricas, sin importar dónde estén almacenadas [30][31]. Entre las características fundamendales que nos provee Grafana están:

- La visualización, muestra múltiples tipos de gráficos que se caracterizan por la rápidez y la flexibilidad con una multitud de opciones.
- Las alertas, permite definir reglas de alerta para las métricas más relevantes. Garantiza su evaluación continuamente y el lanzamiento de alertas vía correo, sms, entre otros.
- Preferencias de usuario, cada usuario podrá configurar según sus gustos los temas (oscuro o blanco), configurar su dashborad por defecto, cambiar las zonas horarias, entre otros.
- Las organizaciones, es posible crear varias organizaciones que permite restringir el acceso en dependencia de los roles. Cada organización contiene sus propias fuentes de datos, paneles y usuarios.

Fuentes de datos

Grafana es capaz de leer las métricas desde diferentes fuentes de datos, tales como:

- InfluxDB
- CloudWatch
- Elasticsearch
- MySQL
- OpenTSBD
- Prometheus
- PostgreSQL
- Graphite

Estas fuentes de datos son las que vienen por defecto, pero es posible incluirles más agregándole pluggins.

Roles

Entre los roles que permite grafana se encuentran:

- Admin, este rol tiene los permisos totales, el mismo puede crear usuarios, dashboard, datasources, entre otros.
- Viewer, este rol contiene solamente los permisos de lectura, solamente los usuario podrán acceder a los dashborad para las organizaciones autorizadas.
- Edit, con este rol los usuarios solamente podrán gestionar dashboard.

Una de las utilidades de Grafana es la autenticación con una tecnología que permita el protoloco LDAP y así restringir el acceso de Grafana con un controlador de dominio.

1.4 Conclusiones parciales.

- Existe gran diversidad de aplicaciones que son utilizadas para la gestión de logs.
- Para satisfacer los requerimientos solicitados por la DTCF en cuanto a la Gestión de Logs es necesario implementar varias herramientas open source.
- Se seleccionan como clientes a implementar: filebeat, winlogbeat y telegraf.
- Graylog es seleccionado como colector de logs y se encarga de su envío indexado hasta Elasticsearch.
- Las métricas serán almacenadas en InfluxDB.
- Grafana será el visor utilizado para graficar los eventos.
- Por el consumo de recursos se recomienda que Graylog, Elasticsearch y Grafana estén instalados en servidores independientes.

Capítulo 2: Sistema de Gestión de logs para la División Territorial de ETECSA en Cienfuegos.

En este capítulo se describe el Sistema de Gestión de Logs implementado en la División Territorial de ETECSA de Cienfuegos como resultado de este trabajo, teniendo en cuenta el diseño y las configuraciones de las herramientas implementadas.

2.1 Diseño del Sistema de Gestión de Logs.

El diseño del Sistema de Gestión de Logs consta de 3 servidores virtuales ubicados en la DMZ entre las redes Corporativa y de Gestión que pertenecen a la DTCF. De esta forma pueden recibir eventos de los diferentes equipos que por cuestiones de seguridad se encuentran en diferentes redes, Figura 1.

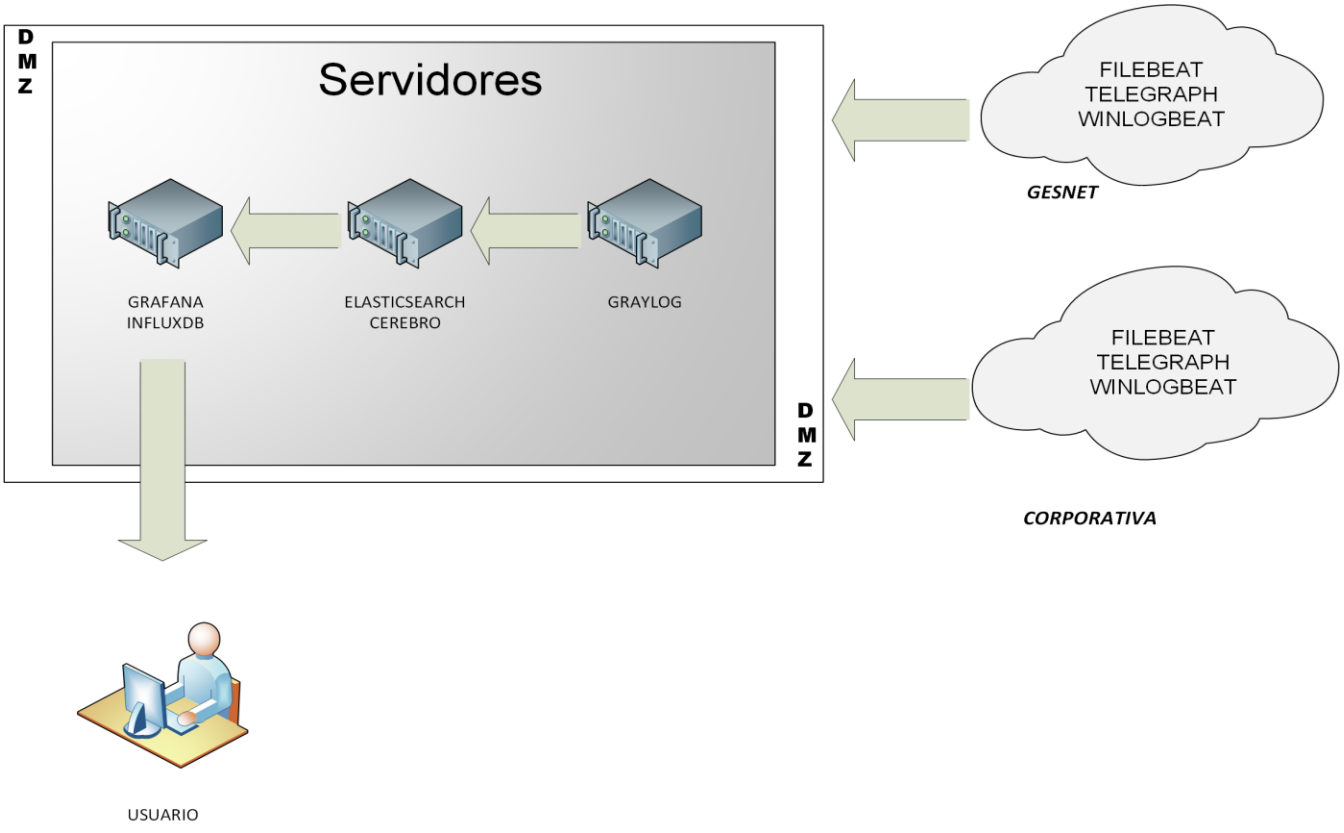

Figura 1 Diseño del Sistema de Gestión de Logs.

Características de los servidores implementados

Servidor Virtual Graylog

- Ubuntu 16.04 x64.
- Java 1.8.3
- Graylog 2.4.3
- MongoDB 3.6.0
- Se le asigna: 4GB de RAM , VCPU (8 cores) y 100GB de HDD.

Servidor Virtual Elasticsearch

- Ubuntu 16.04 x64.
- Java 1.8.3
- Elasticsearch 5.6.2
- curl
- Cerebro 0.6.5
- Se le asigna: 4GB de RAM , VCPU (2 cores) y 100GB de HDD.

Servidor Grafana

- Ubuntu 16.04 x64.
- Java 1.8.3
- Grafana 5.0.4
- InfluxDB 1.5.2
- Se le asigna: 2GB de RAM , VCPU (2 cores) y 15GB de HDD.

En los clientes se instalaron:

- Telegraf 1.6.0
- Filebeat 5.4.0

- Winlogbeat 5.6.2

Durante el diseño es importante tener en cueta las versiones de los softwares por problemas de compatibilidad. Más detalles sobre las versiones compatibles para un diseño similar están disponibles en el Anexo 1.

2.2 Instalación y configuración de Elasticsearch y Cerebro.

Elasticsearch es un popular servidor de búsqueda de código abierto que se utiliza para la búsqueda distribuida en tiempo real y el análisis de datos.

Requisitos para instalación de Elasticsearch en Ubuntu 16.04.

- Java1.8.x
- Curl

Instalación de Elasticsearch

Paso 1: Instalar Java 1.8.X

Se descarga del sitio de Oracle el JDK actual[2] y se procede de la siguiente manera.

El directorio /opt está reservado para todos los paquetes de software y complementos que no forman parte de la instalación predeterminada. Se crea un directorio para la instalación del JDK y extrae java en el directorio:

```
mkdir /opt/jdk

tar -zxf jdk-8u144-linux-x64tar.gz -C /opt/jdk
```

Verificar que el archivo se haya extraído en el directorio /opt/jdk.

```
ls /opt/jdk
```

Configurar la JVM como predeterminada:

```
update-alternatives --install /usr/bin/java java /opt/jre/ jdk-8u144-linux-x64/bin/java 100
```

2 http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

Paso 2: Instalar Elasticsearch:

Instalar el paquete curl para consultar luego el elasticsearch.

```
• apt-get install curl
```

Descargar Elasticsearch[3] e instalarlo.

```
dpkg -i elasticsearch-5.6.2.deb
```

Paso 3: Configurar el servicio de Elasticsearch para que se inicie automáticamente.

```
systemctl daemon-reload
systemctl enable elasticsearch.service
```

Iniciar el servicio.

```
service elasticsearch start
```

Paso 4: Probando Elasticsearch.

Ejecutar desde la terminal lo siguente:

```
curl http://localhost:9200
```

Y se obtiene lo siguiente.

```
{
 "name" : "qNP4INt",
 "cluster_name" : "elasticsearch",
 "cluster_uuid" : "ps78Fq0bRu6jcLGuAPKJMA",
 "version" : {
 "number" : "5.6.2",
 "build_hash" : "57e20f3",
 "build_date" : "2017-09-23T13:16:45.703Z",
 "build_snapshot" : false,
 "lucene_version" : "6.6.1"
```

3 https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-5.6.2.deb

```
 },
 "tagline" : "You Know, for Search"
}
```

Paso 5: Configurar Elasticsearch.

Ejecutar desde la terminal lo siguente:

```
nano /etc/elasticsearch/elasticsearch.yml
```

Editar el fichero elasticsearch.yml y se modifica los aspectos siguientes.

```
#Use a descriptive name for your cluster:
#
 cluster.name: my-application
#Use a descriptive name for the node:
#
 node.name: node-1
#
# Add custom attributes to the node:
#
#node.attr.rack: r1
#Lock the memory on startup:
#
 bootstrap.memory_lock: true
#
#Set the bind address to a specific IP (IPv4 or IPv6):
#
 network.host: IP-my-elastic
#
# Set a custom port for HTTP:
#
#http.port: 9200
```

Ahora es necesario modificar el servicio de Elasticsearch. Ejecutando en el terminal lo siguiente.

```
nano /usr/lib/systemd/system/elasticsearch.service
```

Se descomenta o adiciona la línea siguiente.

```
LimitMEMLOCK=infinity
```

Editar los valores default de Elasticsearch, ejecutando en el terminal.

```
nano /etc/default/elasticsearch
```

Descomentarla línea siguiente:

```
MAX_LOCKED_MEMORY=unlimited
```

y seguidamente para que se establezcan estos cambios al inicio.

```
systemctl daemon-reload
```

Elasticsearch utiliza un directorio de mmapfs de forma predeterminada para almacenar sus índices. Es probable que los límites predeterminados del sistema operativo en los recuentos de mmap sean demasiado bajos, lo que puede provocar excepciones por falta de memoria y se recomienda aumentar la memoria virtual a este valor.

Para esto se edita el fichero sysctl.conf.

```
nano /etc/sysctl.conf
```

Añadiendo lo siguiente:

```
vm.max_map_count=262144
```

Para hacer efectivo el valor agregado se ejecuta:

```
sysctl -p
```

La interacción con el Elasticsearch puede hacerse más amigable a través de *Cerebro* para el cual se puede dedicar una máquina aparte o simplemente instalarlo en el servidor como es el caso.

Instalación y uso de Cerebro

Paso 1: Descargar aplicación [4].

Paso 2: Crear carpeta llamada cerebro en opt.

4 https://github.com/lmenezes/cerebro/releases/download/v0.6.5/cerebro-0.6.5.tgz

```
mkdir /opt/cerebro
```

Y extraer el archivo descargado en la misma.

- tar –zxf cerebro-0.6.5.tgz –C /opt/cerebro

Paso 3: Para ejecutarlo se va directamente a la carpeta bin.

```
cd /opt/cerebro-0.6.5/bin

./cerebro
```

Paso 4: se accede a Cerebro http://IP-Cerebro:9000

Desde esta aplicación se puede acceder a los índices creados y ver el número de archivos que contiene. Además permite realizar acciones sobre éstos en el servidor Elasticsearch Figura 2.

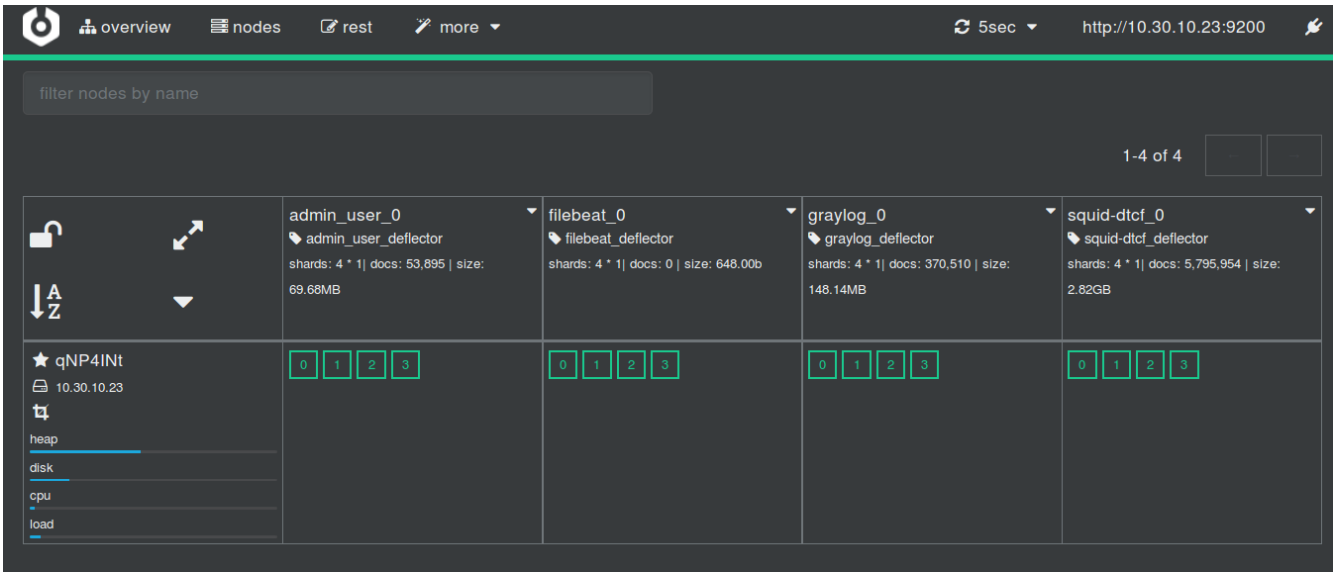

Figura 2. Accediendo al servidor Elasticsearch desde Cerebro.

2.3 Implementación del Graylog y los clientes filebeat.

Es el Graylog la aplicación más compleja que se encuentra durante el proceso de implementación de este diseño. Al ser el encargado de la recolección y clasificación de los logs, los errores en la configuración implicarían pérdida de información a pesar de que todos los servidores se comuniquen entre sí.

La información le llega a través de los clientes filebeat y winlogbeat, por lo que a continuación

se describen primero los clientes y posteriormente los pasos para lograr una configuración efectiva. En cada caso se referenciarán bibliografías que detallan el proceso de instalación de cada herramienta.

2.3.1. Clientes filebeat.

Filebeat es un cargador de datos de registros para archivos locales. Instalado como un agente se encarga de enviar, en este caso, el fichero *auth.log* que contiene los eventos sobre el sistema de autorización de usuarios y permisos en las computadoras con sistemas operativos Linux [32].

Una vez instalado el filebeat [33], se debe configurar el fichero /etc/filebeat/filebeat.yml de la siguiente manera:

```
#============================ Filebeat prospectors ==============================

filebeat.prospectors:
-input_type: syslog
document_type:auth
paths:
 -/var/log/auth.log


#---------------------------- Logstash output --------------------------------
output.logstash:
hosts: ["IP-Servidor-Graylog:5044"]
template.name: "filebeat"
template.path: "filebeat.template.json"
template.overwrite: false

#================================ Logging =====================================
logging.metrics.period: 1m
```

Otros de los objetivos del Filebeat en este diseño es enviar las trazas de Squid al Graylog. Este proceso será descrito en el apéndice 2.3.2.2. En la configuración del fichero *filebeat.yml* antes referida se debe modificar solo los campos:

```
- #============Filebeat prospectors =========

- filebeat.prospectors:
-input_type: log
```

```
- document_type: squid3
- paths:
    -/var/log/squid/*access.log*
```

2.3.2 Instalación y configuración del Graylog.

Graylog es una plataforma de administración de syslog de código abierto que nos ayuda a recopiar, indexar y analizar syslog en una ubicación centralizada. El mismo nos proporciona un sistema unificado y centralizado de mensajes procedentes de diferentes fuentes: sistema operativo, servidores de aplicación, sistemas de información. Dispone de un sistema de alertas y de búsqueda de histórico de logs usando ElasticSearch como base de datos de índices [34].

Paso 1: Instalar las herramientas y dependencias necesarias

```
apt install apt-transport-https openjdk-8-jre-headless uuid-runtime pwgen mc htop libdigest-sha-perl
```

Paso 2: Como Graylog usa MongoDB como base de datos , se procede a instalarlo primero.

Agregar la llave pública de su repositorio para instalar paquetes firmados.

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 –recv
2930ADAE8CAF5059EE73BB4B58712A2291FA4AD5 (sin proxy)

sudo apt-key adv --keyserver-options http-proxy=http://ip_proxy:port/ --keyserver
hkp://keyserver.ubuntu.com:80 --recv 2930ADAE8CAF5059EE73BB4B58712A2291FA4AD5 (con
proxy)
```

Agregar el repositorio de MongoDB.

```
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.6
multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-3.6.list
```

Actualizar.

```
1. root@graylog:~# apt update
2. Hit:1 http://archive.ubuntu.com/ubuntu xenial InRelease
3. Get:2 http://archive.ubuntu.com/ubuntu xenial-updates InRelease [102 kB]
4. Get:3 http://archive.ubuntu.com/ubuntu xenial-security InRelease [102 kB]
```

```
5. Ign:4 https://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.6 InRelease
6. Get:5 https://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.6 Release [3457 B]
7. Get:6 https://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.6 Release.gpg [801 B]
8. Get:7 https://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.6/multiverse amd64
   Packages [4058 B]
9. Get:8 https://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.6/multiverse arm64
   Packages [4044 B]
10. Fetched 217 kB in 3s (69.7 kB/s)
11. Reading package lists... Done
12. Building dependency tree
13. Reading state information... Done
14. All packages are up to date.
15. root@graylog:~#
```

Instalar MongoDB

```
apt install -y mongodb-org
```

Iniciar servicio.

```
sudo systemctl start mongod
```

Hacer autoarrancable el servicio.

```
root@graylog:~# sudo systemctl enable mongod

Created symlink from /etc/systemd/system/multi-user.target.wants/mongod.service to
/lib/systemd/system/mongod.service

root@graylog:~#
```

Paso 3: Agregar repositorio de Graylog al listado de fuentes.

```
wget https://packages.graylog2.org/repo/packages/graylog-2.4-repository_latest.deb
```

Instalar.

```
sudo dpkg -i graylog-2.4-repository_latest.deb
```

Actualizar e instalar Graylog.

```
sudo apt update
```

```
sudo apt install graylog-server
```

Configurar la clave de admin(root) y la clave secreta, ambas serán usadas para levantar el servidor GrayLog.

```
root@graylog:~# SECRET=$(pwgen -s 96 1)

root@graylog:~# cp /etc/graylog/server/server.conf /etc/graylog/server/server.conf.orig

root@graylog:~# sudo -E sed -i -e 's/password_secret =.*/password_secret = '$SECRET'/'
/etc/graylog/server/server.conf
```

La llave secreta quedará directamente registrada en la configuración del servidor GrayLog. Ahora se pasa la clave de admin.

```
1. root@graylog:~# PASSWORD=$(echo -n password | shasum -a 256 | awk '{print $1}')
2. root@graylog:~# sudo -E sed -i -e 's/root_password_sha2 =.*/root_password_sha2 = '$PASSWORD'/'
   /etc/graylog/server/server.conf
3. root@graylog:~#
```

Donde *password* es la clave asignada al usuario *admin* del Graylog y quedará directamente reflejado en su configuración.

Editar la configuración del servidor y asegurarse de ver que ambas claves se insertaron bien

```
nano /etc/graylog/server/server.conf
```

Configurar otros aspectos de interés

```
# The time zone setting of the root user. See http://www.joda.org/joda-time/timezones.html for a list of valid time zones.
# Default is UTC
root_timezone = America/Havana

# REST API listen URI. Must be reachable by other Graylog server nodes if you run a cluster.
# When using Graylog Collectors, this URI will be used to receive heartbeat messages and must be accessible for all collectors.
rest_listen_uri = http://0.0.0.0:9000/api/

# Web interface listen URI.
# Configuring a path for the URI here effectively prefixes all URIs in the web interface. This is a replacement
```

```
# for the application.context configuration parameter in pre-2.0 versions of the Graylog web interface.
web_listen_uri = http://0.0.0.0:9000/

# List of Elasticsearch hosts Graylog should connect to.
# Need to be specified as a comma-separated list of valid URIs for the http ports of your elasticsearch nodes.
# If one or more of your elasticsearch hosts require authentication, include the credentials in each node URI that
# requires authentication.
#
# Default: http://127.0.0.1:9200
elasticsearch_hosts = http://IP-Elastic:9200/

# How many Elasticsearch shards and replicas should be used per index? Note that this only applies to newly created indices.
# ATTENTION: These settings have been moved to the database in Graylog 2.2.0. When you upgrade, make sure to set these
# to your previous settings so they will be migrated to the database!
elasticsearch_shards = 1
elasticsearch_replicas = 0

# The email address of the root user.
# Default is empty
root_email = correo_alertas@dominio

# REST API transport address. Defaults to the value of rest_listen_uri. Exception: If rest_listen_uri
# is set to a wildcard IP address (0.0.0.0) the first non-loopback IPv4 system address is used.
# If set, this will be promoted in the cluster discovery APIs, so other nodes may try to connect on
# this address and it is used to generate URLs addressing entities in the REST API. (see rest_listen_uri)
# You will need to define this, if your Graylog server is running behind a HTTP proxy that is rewriting
# the scheme, host name or URI.
# This must not contain a wildcard address (0.0.0.0).
rest_transport_uri = http://IP-Graylog:9000/api/
```

Hacer autoarrancable nuestro servicio:

```
sudo systemctl enable graylog-server.service
```

Iniciar el servicio:

```
1. sudo systemctl start graylog-server.service
```

Ya se puede acceder al Graylog (http://IP-Graylog:9000) con las credenciales *admin* y el *password* definido anteriormente.

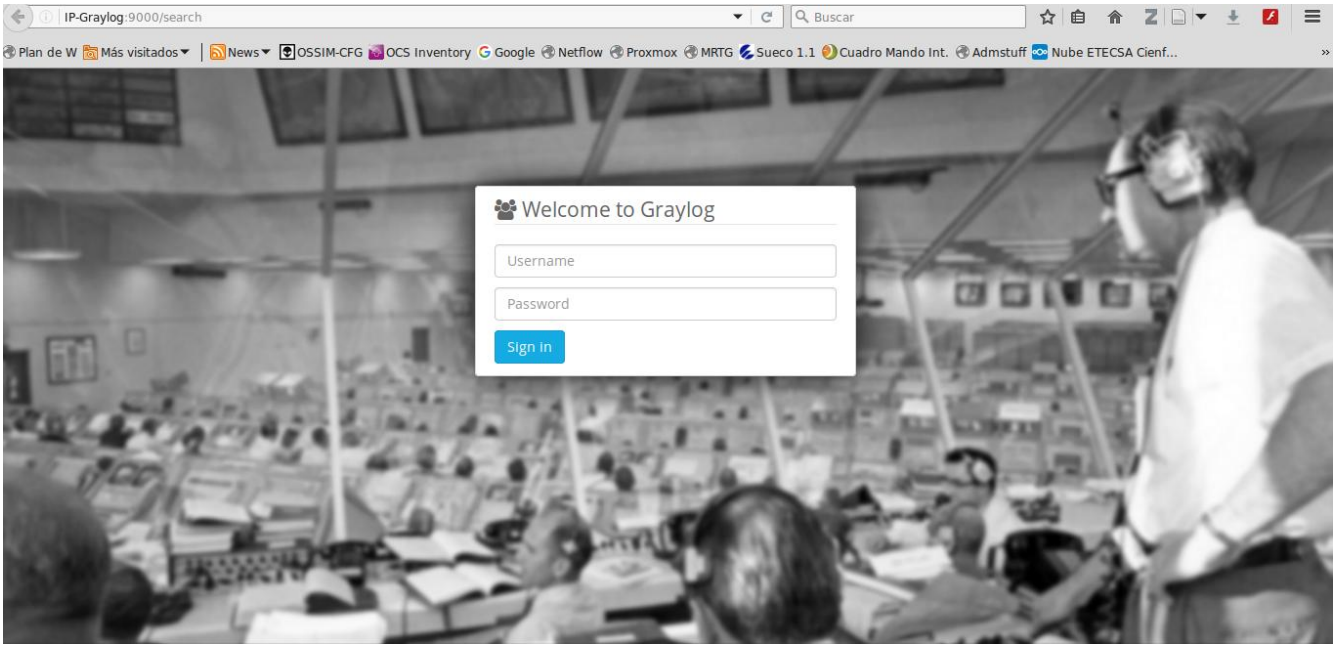

Figura 4. Interfaz de Graylog

2.3.2.1 Configuraciones iniciales.

Lo primero es hacer ajustes en la configuración de Graylog [35].

Procesador de geolocalización

Por defecto Graylog no trae una base datos para la geo localizacion por lo que es necesrio descargarla de MaxMind City databases[5] .

```
wget -t0 -c http://geolite.maxmind.com/download/geoip/database/GeoLite2-City.tar.gz
```

Seguidamente se extrae el contenido del fichero y se copia el fichero GeoLite2-City.mmdb a /etc/graylog/server .

```
tar -xvf GeoLite2-City.tar.gz

cp GeoLite2-City_20180306/GeoLite2-City.mmdb /etc/graylog/server
```

Ahora en la web de Graylog hay que activar el Geo-Location Processor (system/Configuration) marcando en *Enable Geo-Location porcessor* lo demás se deja por

5 *MaxMind* es un proveedor líder de inteligencia de IP y herramientas de prevención de fraude en línea.

defecto. En la medida que vengan mensajes que contengan ips los procesará obteniendo datos adicionales como las coordenadas, código pais, nombre de ciudad.

Configuración de procesadores de mensajes.

Es importante tener un orden de procesamiento que nos permita obtener lo que se desea a la hora de parsear los mensajes de logs obtenidos desde los *inputs*.

En *system/Configuration/Message Processors Configuration* debe quedar establecido el siguiente orden con todos habilitados:

- AWS Instance Name Lookup.
- Message Filter Chain que procesa los inputs.
- Pipeline Procesos si se quiere tener a los campos estáticos creados por los extractores.
- Geo-Location Processor para procesar las ips descubiertas agregando campos asociados a la misma como localizacion geográfica, codigo del pais y nombre la de ciudad.

2.3.2.2 Definiendo Inputs y Streams.

Aunque el Graylog está diseñado para tener diferentes *inputs*[6] solo se implementaron las de tipo *Beats* donde los clientes filebeat y winlogbeat envían los datos al servidor por el puerto 5044.

6 Entradas de datos.

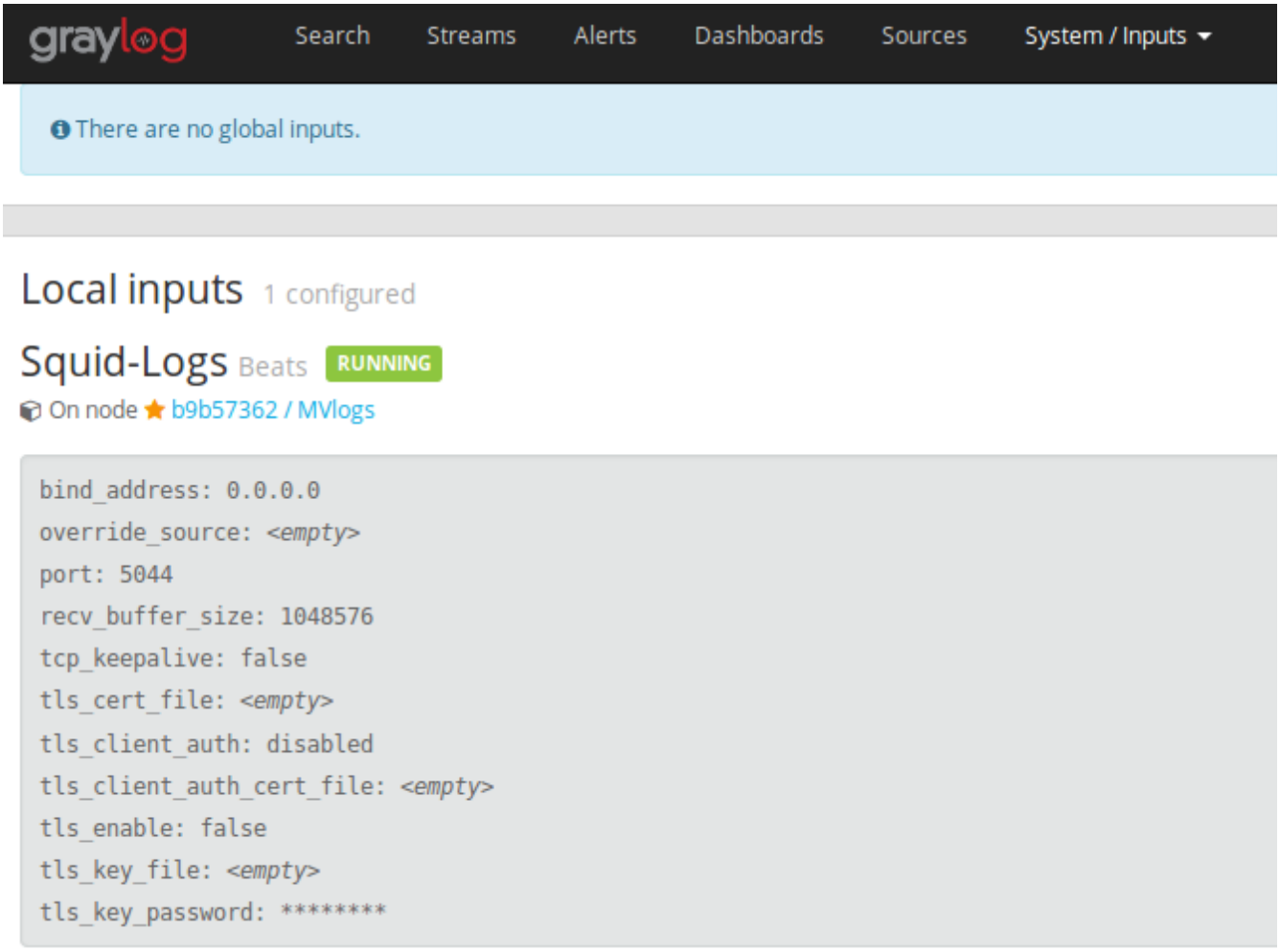

Figura 5. Input de tipo Beats para recibir datos por el puerto 5044 desde cualquier IP.

Los inputs se pueden definir manualmente o importando *Content Packs*[7] ya generados disponibles en Internet que incluye Input, extractores, patrones y por último el stream para que el input lo alamacene en índice creado.

En el caso de los logs Squid se utilizó un Content Pack[8] disponible gratis en Internet y para los logs de autenticación se utilizó este input pero fue necesario definir los índices, patrones Grok y extractors personalizados.

Los *streams* en Graylog permiten enrutar los mensajes por categoría en tiempo real mientras

7 Un **paquete de contenido** es un tipo de archivos que incluye contenidos y metadatos, que en el caso de Graylog incluye lo que necesitas para trabajar con determinados logs.

8 Disponible en https://marketplace.graylog.org/addons?tag=Squid

son procesados. En este caso se tiene que para el input definido solo los que cumplen alguna de las siguientes condiciones:

- El campo *type* contiene *squid3*
- El campo *type* contiene *auth*
- El campo *facility* contiene *winlogbeat*

Las dos primeras condiciones se cumplen cuando en el fichero de configuración de los clientes filebeat se define el campo *document_type*. Los clientes winlogbeat lo reflejan en al campo *facility*.

2.3.2.3 Índices.

Graylog tiene su propio índice que genera por defecto que es graylog. En system/Indices se puede adicionar los propios que en este caso son para los logs de autenticación y los de Squid. En este caso se define que deben rotar mensual y que elimine los indices pasados los 12 meses.

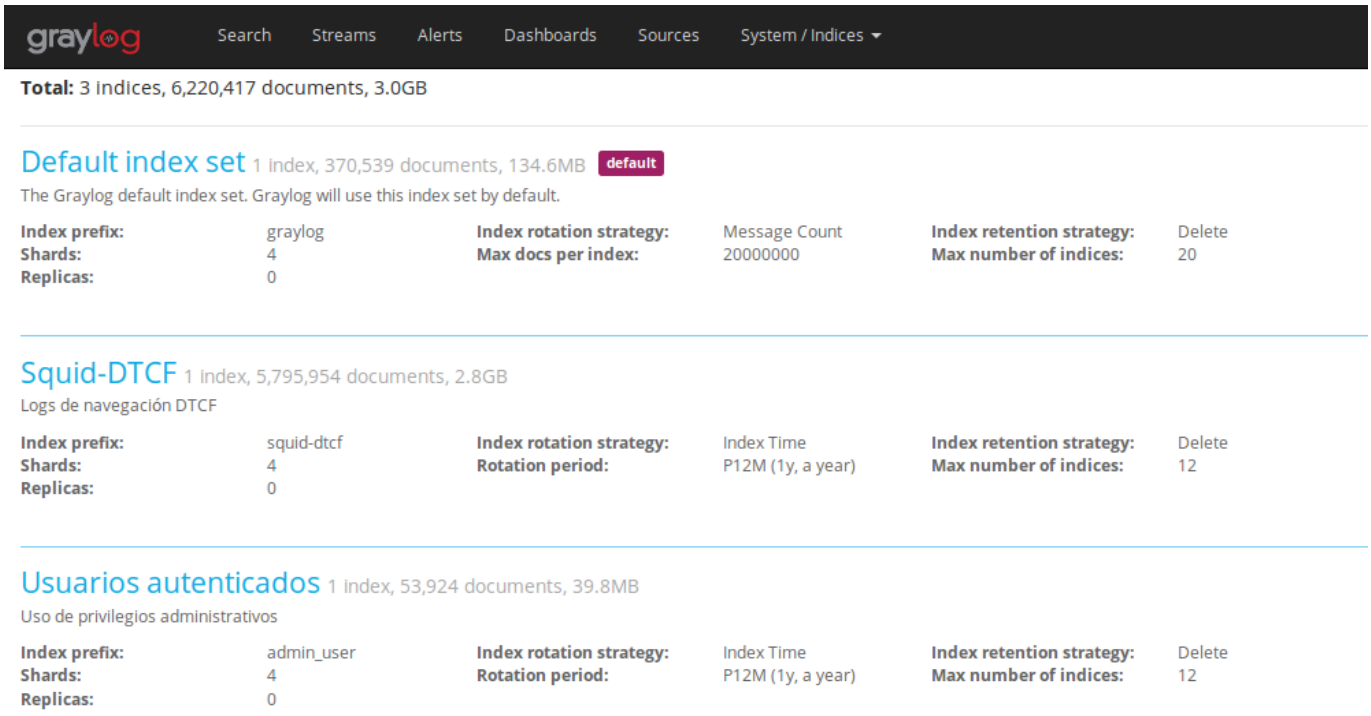

Figura 6. Índices definidos.

Estos índices se crean de manera inmediata y con *Cerebro* se puede comprobarlo.

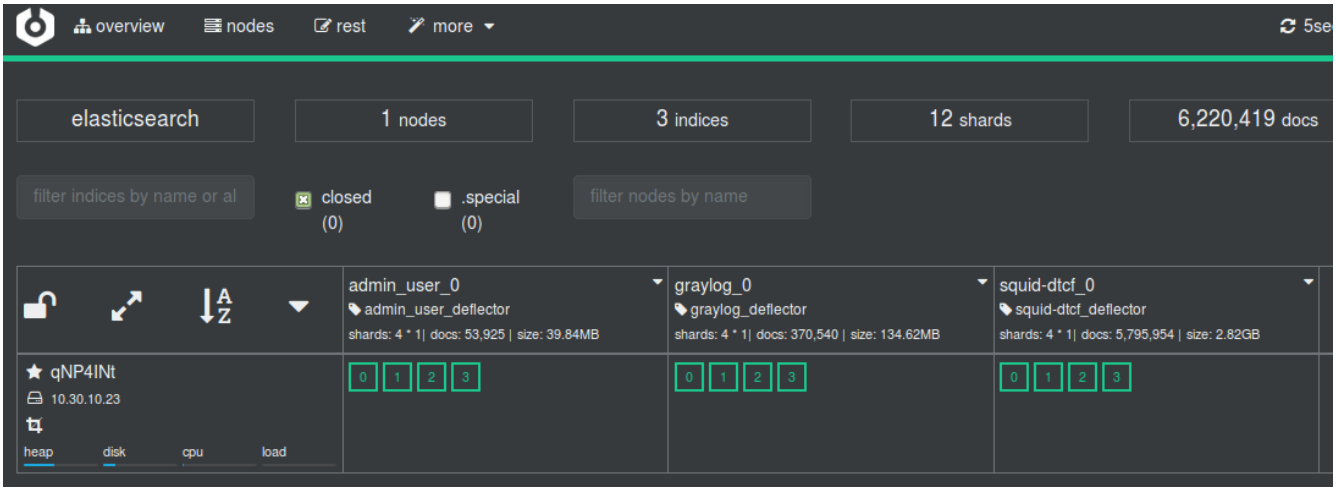

Figura 7. Comprobando que los índices creados en Graylog están en el Elasticsearch.

2.3.2.4 Manejo de extractors a partir de patrones Grok y expresiones regulares.

Los patrones Grok[9] son especializados en deserialización de archivos de texto sin estructurar, normalmente logs. Cada patrón de Grok es una expresión regular con nombre. Pueden identificarse y volverse a utilizar estos patrones de deserialización según sea necesario. De ese modo resulta más sencillo utilizar Grok en comparación con el uso de expresiones regulares. Grok proporciona un conjunto de patrones predefinidos pero también pueden crearse patrones personalizados [36].

Es la confección de estos patrones personalizados para las necesidades de la DTCF un aporte de este trabajo que facilitaría la implementación de este diseño en otros escenarios.

Los extractores le permiten instruir a los nodos Graylog sobre cómo extraer datos de cualquier texto en el mensaje recibido [37].

En Graylog se puede importar o crear patrones Grok accediendo a la opción de menú *System/Grok Patterns* Figura 8. Éstos serán posteriormente referenciados en el proceso de crear los *extractors.*

9 Grok se traduce como “entender algo intuitivamente” por lo que Grok pattern se entiende como “patrón que se entiende de manera intuitiva”

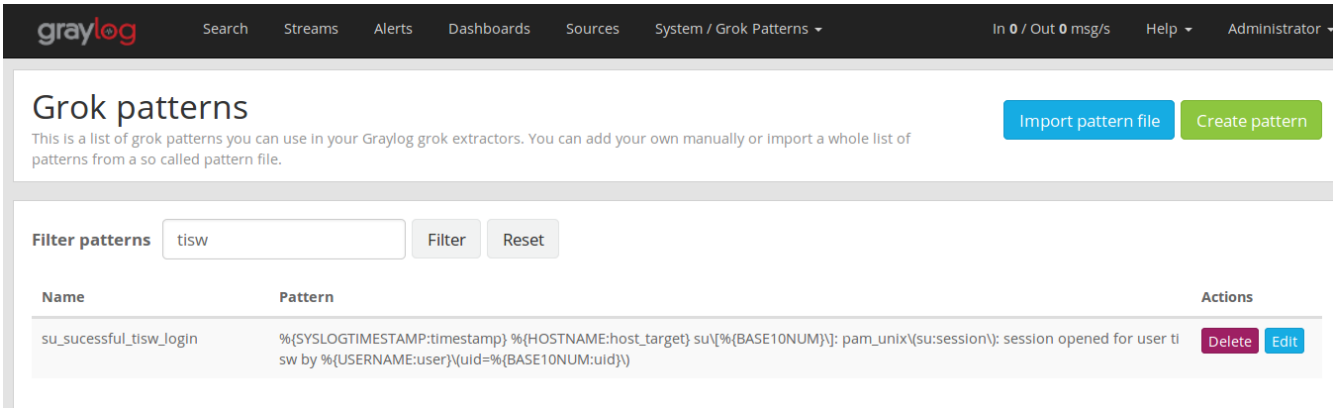

Figura 8. Ejemplo de Grok pattern para autenticación con éxito del usuario “tisw”.

Los *extractors* se definen para las *inputs,* en este caso la *input* de tipo *Beats* es *Squid-Logs* y se le asocian los Groks importados del Content Pack para Squid y los definidos para los diferentes eventos que pertenecen a los requisitos funcionales definidos por la DTCF.

A continuación se muestran los Groks utilizados para definir los extractors de eventos de autenticación y el Content Pack para Squid puede consultarse en el Anexo 2.

Tabla 1. Patrones Grok definidos.

Nombre del patrón Grok	Evento	Patrón Grok

ssh_brute_force_attack	Failed password for invalid user	grok {
		type => "linux-syslog"
		pattern => "%{SYSLOGTIMESTAMP:timestamp} %{HOSTNAME:host_target} sshd\[%{BASE10NUM}\]: Failed password for invalid user %{USERNAME:username} from %{IP:src_ip} port %{BASE10NUM:port} ssh2"
		add_tag => "ssh_brute_force_attack"
		}

ssh_failed_login	Failed password for <usuario> from <IP> port <puerto>	grok {
		type => "linux-syslog"
		pattern => "%{SYSLOGTIMESTAMP:timestamp} %{HOSTNAME:host_target} sshd\[%{BASE10NUM}\]: Failed password for %{USERNAME:username} from %{IP:src_ip} port %{BASE10NUM:port} ssh2"
		add_tag => "ssh_failed_login"
		}

Nombre del patrón Grok	Evento	Patrón Grok

ssh_sucessful_login	Accepted password for <usuario> from <IP> port <puerto>	grok {
		type => "linux-syslog"
		pattern => "%{SYSLOGTIMESTAMP:timestamp} %{HOSTNAME:host_target} sshd\[%{BASE10NUM}\]: Accepted password for %{USERNAME:username} from %{IP:src_ip} port %{BASE10NUM:port} ssh2"
		add_tag => "ssh_sucessful_login"
		}

sudo_auth_failure	sudo: pam_unix(sudo:auth): authentication failure;	grok {
		type => "linux-syslog"
		pattern => "%{SYSLOGTIMESTAMP:timestamp} %{HOSTNAME:host_target} sudo: pam_unix\(sudo:auth\): authentication failure; logname=%{USERNAME:logname} uid=%{BASE10NUM:uid} euid=%{BASE10NUM:euid} tty=%{TTY:tty} ruser=%{USERNAME:ruser} rhost=(?:%{HOSTNAME:remote_host}|\s*) user=%{USERNAME:user}"
		add_tag => "sudo_auth_failure"
		}

su_sucessful_root_login	Successful su for root by root	grok {
		type => "linux-syslog"
		pattern => %{SYSLOGTIMESTAMP:timestamp} %{HOSTNAME:host_target} su\[%{BASE10NUM}\]: Successful su for root by root
		add_tag => "su_sucessful_root_login"
		}

sudo_sucessful_as_tisw	session opened for user root by <user>	grok {
		type => "linux-syslog"
		pattern => %{SYSLOGTIMESTAMP:timestamp} %{HOSTNAME:host_target} su\[%{BASE10NUM}\]: pam_unix\(su:session\): session opened for user tisw by %{USERNAME:user}\(uid=%{BASE10NUM:uid}\)
		add_tag => "sudo_sucessful_as_root"
		}

Nombre del patrón Grok	Evento	Patrón Grok

sudo_sucessful_as_root	session opened for user root by <user>	grok {
		type => "linux-syslog"
		pattern => %{SYSLOGTIMESTAMP:timestamp} %{HOSTNAME:host_target} su\[%{BASE10NUM}\]: pam_unix\(su:session\): session opened for user root by %{USERNAME:user}\(uid=%{BASE10NUM:uid}\)
		add_tag => "sudo_sucessful_as_root"
		}

su_sucessful_tisw_login	Successful su for tisw by <usuario>	grok {
		type => "linux-syslog"
		pattern => %{SYSLOGTIMESTAMP:timestamp} %{HOSTNAME:host_target} su\[%{BASE10NUM}\]: Successful su for tisw by %{USERNAME:username}
		add_tag => "su_sucessful_tisw_login"
		}

El listado de extractors para el inputs Squid-Logs está disponible en el Anexo 3.

2.4 InfluxDB y telegraf para recolección de métricas.

Una de las principales tareas que tiene los administradores de red es la de monitorizar la infraestructura, de la manera más automatizada posible. Hay varias partes en el monitoreo, uno de ellos es el almacenamiento de las métricas, que son las variables contables generadas por los servidores.

Una excelente combinación de productos para monitoreo de métricas es: InfluxDB, Telegraf y Grafana. Entre otras ventajas tiene que con posterioridad se pueden seguir adicionando clientes de diferentes sistemas operativos.

2.4.1 Instalación y configuración de InfluxDB.

Para instalar InfluxDB se añade el repositorio [38].

```
curl -sL https://repos.influxdata.com/influxdb.key | sudo apt-key add –
source /etc/lsb-release
echo "deb https://repos.influxdata.com/${DISTRIB_ID,,} ${DISTRIB_CODENAME} stable" | sudo tee
        /etc/apt/sources.list.d/influxdb.list
```

Y se procede con la instalación.

```
sudo apt-get update && sudo apt-get install influxdb
```

Una vez terminada la instalación se inicia el servicio.

```
/etc/init.d/influxdb restart
```

Al ya tener el repositorio de influxdata añadido, será sencillo instalar el Telegraf.

```
sudo apt-get install telegraf
```

Es necesario editar la configuración de Telegraf ubicado en /etc/telegraf.

```
vi /etc/telegraf/telegraf.conf
```

Añadirendo la URL del servidor InfluxDB en la sección llamada outputs.influxdb.

```
1.  [[outputs.influxdb]]
2.  ## The full HTTP or UDP endpoint URL for your InfluxDB instance.
3.  ## Multiple urls can be specified as part of the same cluster,
4.  ## this means that only ONE of the urls will be written to each interval.
5.  # urls = ["udp://localhost:8089"] # UDP endpoint example
6.  urls = ["http://IP-InfluxDB:8086"]
7.  ## The target database for metrics (telegraf will create it if not exists).
8.  database = "telegraf"
9.  ## Precision of writes, valid values are "ns", "us" (or "¬µs"), "ms", "s", "m", "h".
10. ## note: using "s" precision greatly improves InfluxDB compression.
11. precision = "s"
12. ## Retention policy to write to.
13. retention_policy = "default"
14. ## Write consistency (clusters only), can be: "any", "one", "quorom", "all"
15. write_consistency = "any"
16. ## Write timeout (for the InfluxDB client), formatted as a string.
17. ## If not provided, will default to 5s. 0s means no timeout (not recommended).
18. timeout = "5s"
```

Se reinicia el servicio de Telegraf.

```
/etc/init.d/telegraf restart
```

Hecho esto ya el servidor está listo para recolectar las métricas.

2.4.2 Clientes Telegraf.

Telegraf es un agente de código abierto para recopilar métricas y datos en el sistema en el que se ejecuta o desde otros servicios. Puede utilizarse en diferentes sistemas operativos [39].

El proceso de instalación fue descrito en el apéndice anterior y se aplica al resto de los clientes que se desea monitorear.

En el fichero *telegraf.conf* se definen además lo que se desea monitorear. A continuación puede verse un fragmento de este fichero dedicado a los parámetros de la CPU.

```
###############################################################################
# INPUT PLUGINS #
###############################################################################

# Read metrics about cpu usage
 [[inputs.cpu]]
 ## Whether to report per-cpu stats or not
 percpu = true
 ## Whether to report total system cpu stats or not
 totalcpu = true
 ## If true, collect raw CPU time metrics.
 collect_cpu_time = false
 ## If true, compute and report the sum of all non-idle CPU states.
```

```
report_active = true
```

Por defecto, la configuración de Telegraf no viene con el plugin de networking habilitado, pero es tan sencillo como buscar por inputs.net y descomentar la línea.

2.5 Grafana como visor de eventos.

Grafana es el dashboard que se encarga de mostrar toda la información, ya sea de InfluxDB o Elasticsearch, aunque permite otras entradas de datos.

Instalando y configurando Grafana

Para poder instalar Grafana será necesario añadir los repositorios oficiales.

```
echo "deb https://packagecloud.io/grafana/stable/debian/ wheezy main" | sudo tee
/etc/apt/sources.list.d/grafana.list
curl https://packagecloud.io/gpg.key | sudo apt-key add -
sudo apt-get install -y apt-transport-https
sudo apt-get update
```

Realizar la instalación de la siguiente manera.

```
sudo apt-get install grafana
```

Una vez terminado de instalar, se procede a reiniciar el servicio.

```
sudo /etc/init.d/grafana-server start
```

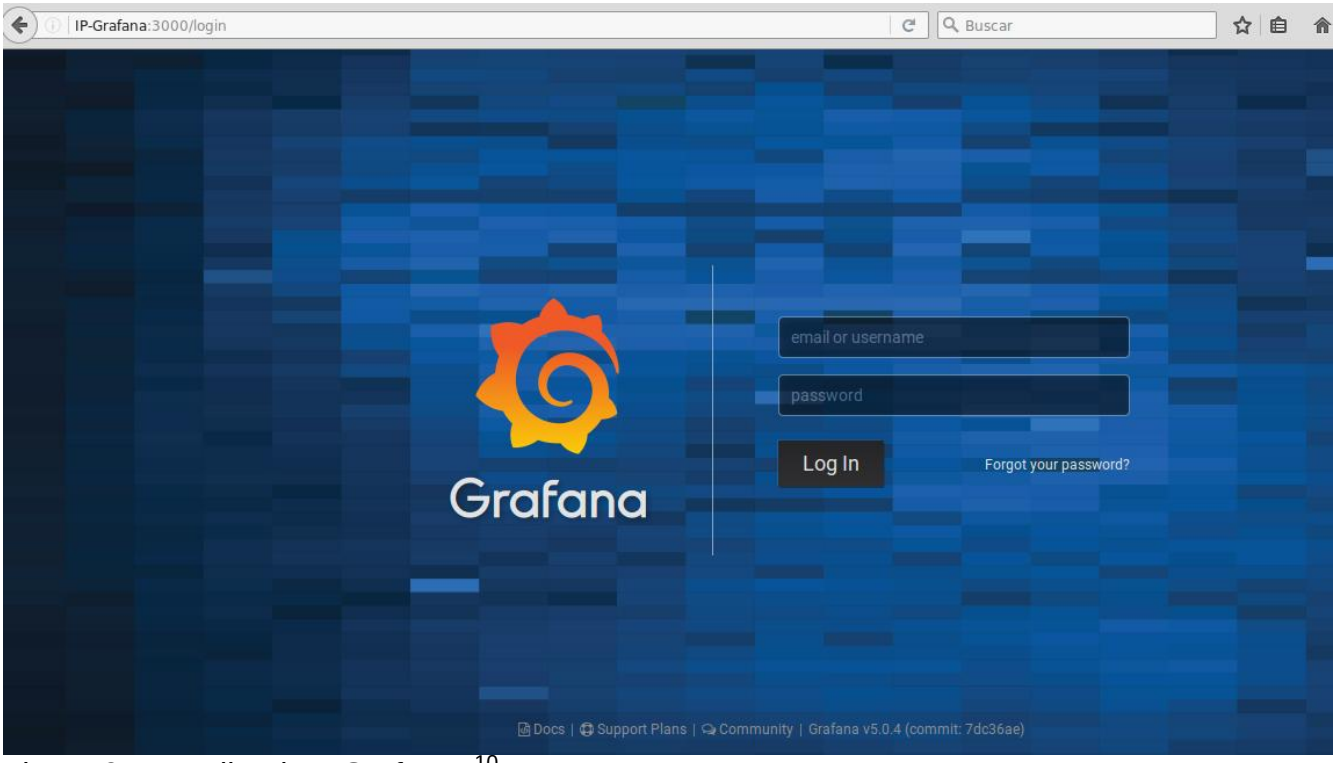

Figura 9. Accediendo a Grafana. [10]

Por defecto Grafana trae unos cuantos plugins pero serán necesario instalar otros de gran utilidad para los Dashboard disponibles en http://grafana.com/plugins?utm_source=grafana_plugins_list. En este caso se utiliza los plugins de panel.

Definiendo los Data Source

Para cada fuente de datos se debe crear un Data Source y en el caso de Elasticsearch uno por cada índice que se desea graficar. Accediendo a la opción de menú *Configuration/ Data Sources/ +Add data source*

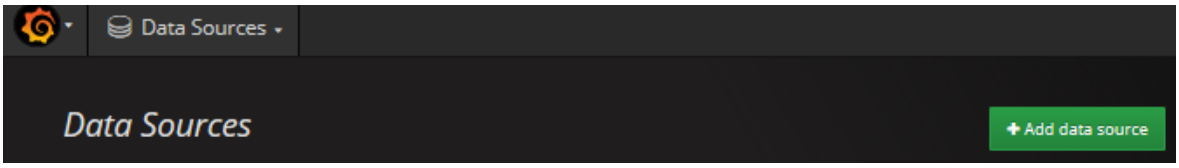

Figura 10. Adicionar Data source.

[10] http//IP-Grafana:3000

Se crean en este caso tres Data Source:

- Squid-graylog
- Auth-logs
- MetricsInfluxDB

El tipo de conexión es Elasticsearch o InfluxDB según corresponda con las url de los servidores correspondientes con el puerto 9200 para Elastic y 8086 para InfluexDB.

En el caso de *MetricsInfluxDB* los detalles de la conexión quedaron definidos como se observa en la siguiente Figura.

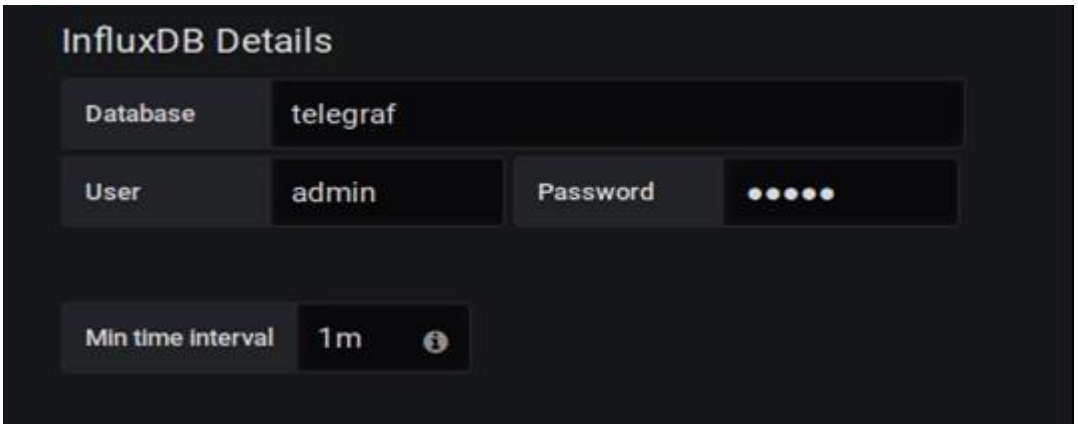

Figura 11. Configuración de InfluxDB.

Detalles de la configuración del Data Source MetricsInfluxDB Squid-graylog y Auth-logs quedaron definidos de forma similar pero con la diferencia del nombre del índice, Figura 12.

Figura 12. Detalles de la configuración del Data Source Squid-graylog

Importando Dashboard

Accediendo a la opción de menú *Create/Import* abre un diálogo en el cual se puede subir el fichero .json correspondiente Figura 13. En *Grafana Dasboard*[11] existen algunos públicos de los que se utilizaron para *Squid* y *Telegraf*.

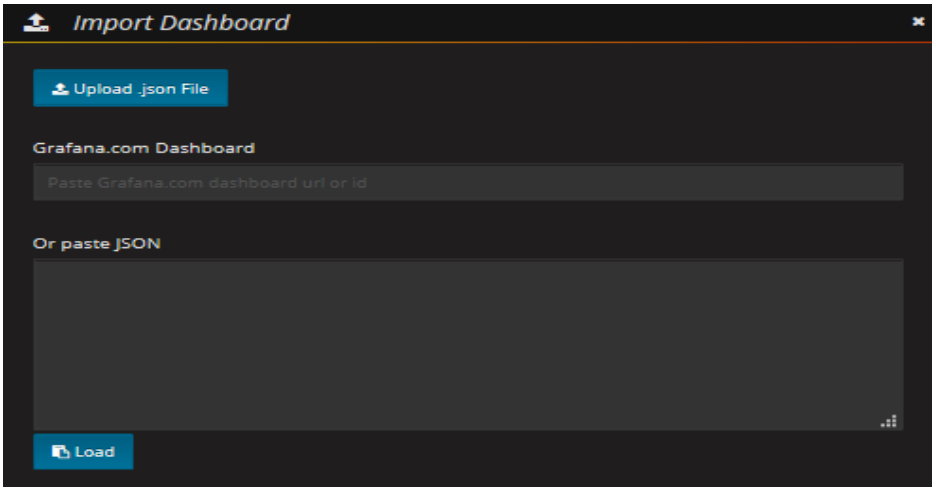

Figura 13. Importando Dashboard

2.5.1 Monitoreo de trazas de Proxy Squid.

Se sube el fichero descargado **Upload .json file** y lo que se asocia al datasource Squid-Graylog, Figura 14.

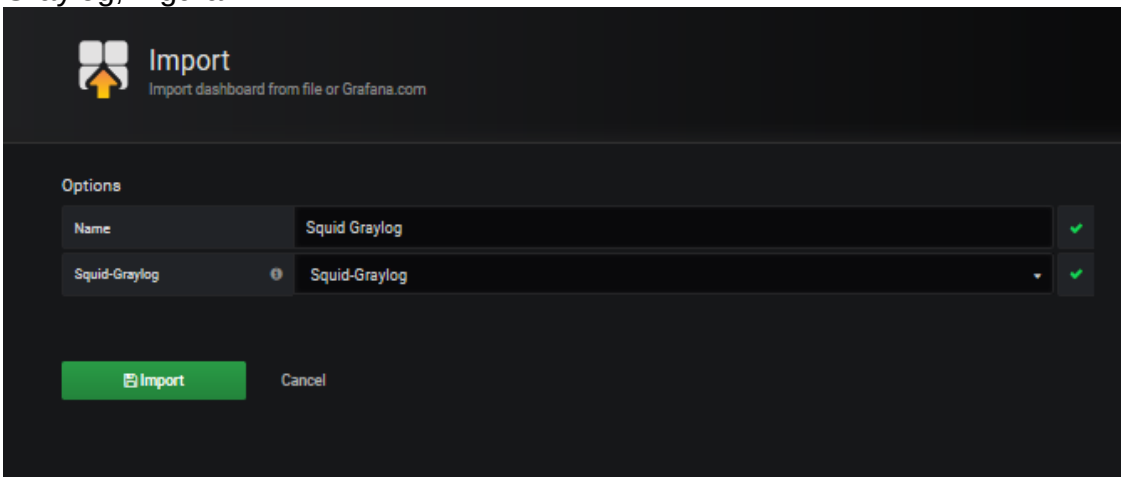

Figura 14. Importando dashboard Squid y asignado el Data source Squid-Graylog.

De esta forma se puede ver el dashboard funcionando Figura 15.

11 *https://grafana.com/dashboards*

Figura 15. Dashboard de Squid graficando resultados.

El uso de este Dashboard tiene en la DTCF características especiales, porque el proxy no es administrado en el territorio y solo se puede acceder al fichero de trazas del día anterior que es voluminoso.

Fue necesario un script que automatizara la descarga del fichero y la posterior ejecución del cliente filebeat para el envío de los datos. Esto permite tener la información actualizada según el flujo de publicación de los logs sin intervención de terceros.

2.5.2 Métricas en los servidores.

El monitoreo del compartamiento de las métricas de los servidores en una red es de mucha importancia para garantizar el funcionamiento de los servicios que prestan y poder tomar decisiones oportunas de balance de carga antes de que ocurran interrupciones. En este proceso la generación de alarmas que permitan avisos por correo, sms, etc ayudan a que el personal de Tecnologías de la Información no esté necesariamente monitoreando el Dashboard de Grafana para tal efecto.

Son los administradores de red los encargados de definir los umbrales que en las métricas se considerarían alarmantes, en este caso fueron:

- Memoria RAM por más de 1h al 95%.
- CPU ocupada por más de 1h al 95%.
- Almacenamiento ocupado en un 90%.

Un aspecto útil es que no son umbrales fijos, el cambio en la configuración de éstos en Grafana es un trabajo simple y a partir de experiencias prácticas son personalizables.

2.5.2.1 Alarmas.

Cuando es de interés programar alarmas, Grafana cuenta con un sistema de alerta que controla y te notifica cuando se activa una de éstas.

A partir de un Panel donde se grafica una métrica se puede definir la alarma como se muestra en la Figura 16. Para ello se establecen condiciones de tiempo y valores basados en estadísticas. Además permite personalizar el mensaje, Anexo 4.

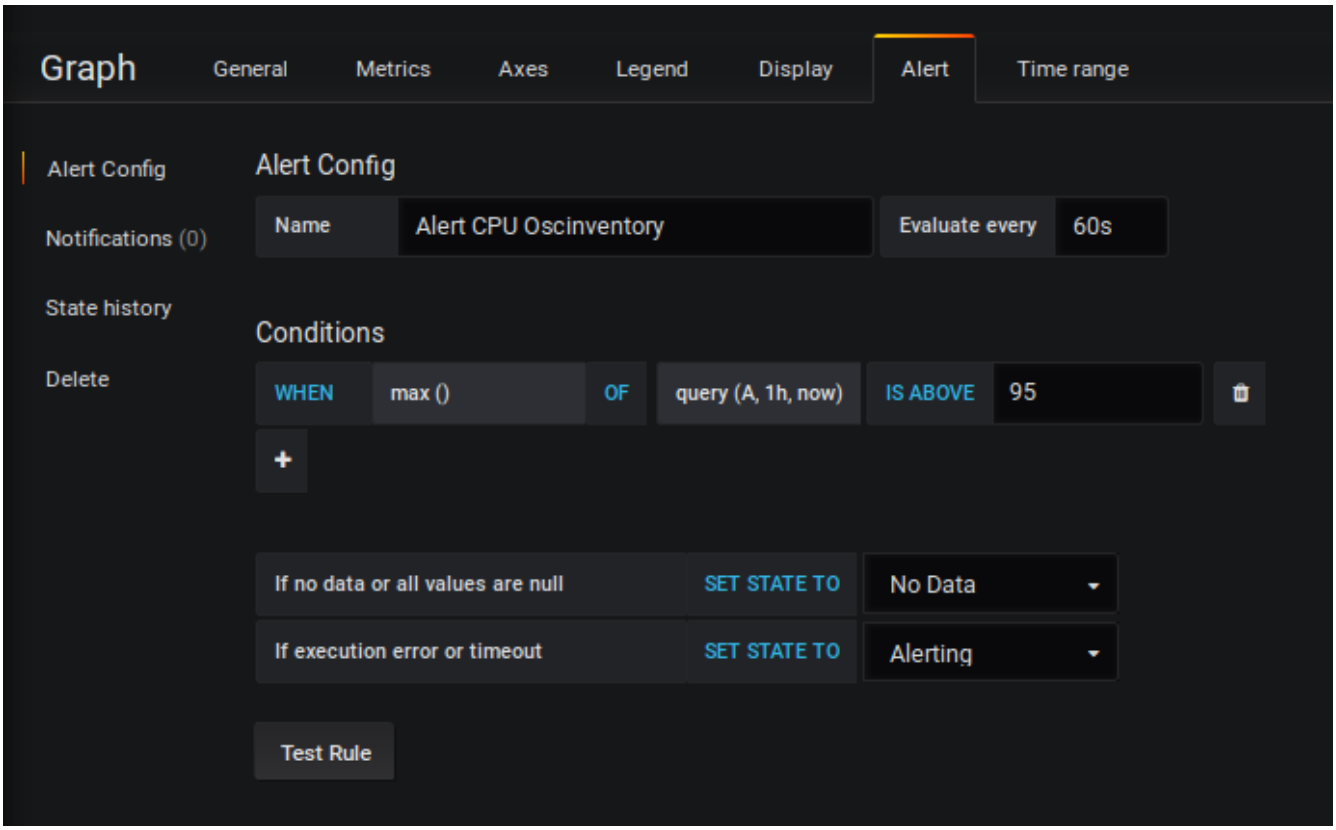

Figura 16. Definiendo alarmas.

2.5.3 Gestión de eventos de autenticación.

El mayor grado de personalización de este trabajo pertenece a la gestión de eventos de autenticación como se pudo apreciar en la Tabla 1, apéndice 2.3.2.4..

Para esta gestión se realizó un Dashboard donde pueden filtrarse los eventos por origen y tipo.

2.5.3.1 SSH

Se analizan los eventos ssh de autenticación fallida y exitosa, así como posibles ataques por fuerza bruta al servicio.

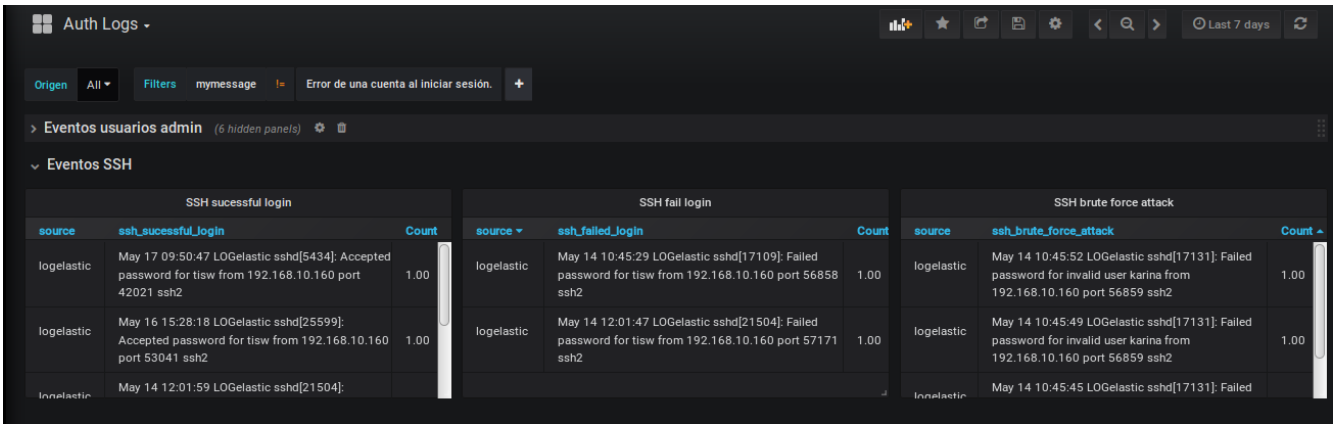

Figura 17. Dashboard para eventos de autenticación donde se muestran los que pertenecen a SSH.

2.5.3.2 Usuarios administradores.

La gestión de usuarios que hacen uso de privilegios administrativos se separa en varias categorías relacionadas con:

- Autenticarse como administrador local.
- Adición de nuevo usuario.
- Modificación de privilegios.
- Escalada de privilegios.

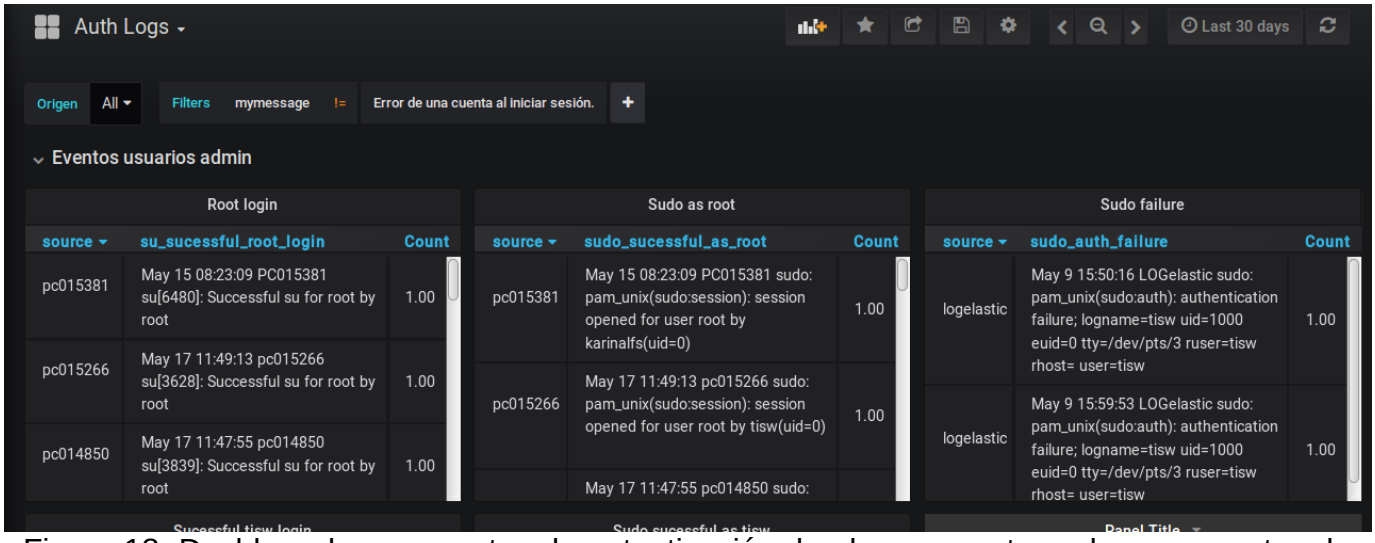

Figura 18. Dashboard para eventos de autenticación donde se muestran algunos eventos de usuarios.

2.6 Conclusiones parciales.

- Se realizó un Diseño de Sistema de Gestión de Logs basado en software libre que responde a las necesidades de la DTCF.
- Las herramientas seleccionadas permiten continuar incorporando nuevos elementos de monitoreo.
-

Capítulo 3: Evaluación del Sistema de Gestión de Logs mediante la simulación de eventos y análisis de los resultados.

Se realiza una evaluación del sistema propuesto a partir de crear los escenarios necesarios que generaran logs que es interés de la Entidad no pasen inadvertidos.

3.1 Método de evaluación.

El Sistema de Gestión de Logs implementado en la DTCF tiene tres áreas objeto de evaluación:

- Métricas de los servidores.
- Trazas de Squid Proxy.
- Eventos asociados a la autenticación de usuario que fueron definidos como requerimientos.

La evaluación de este sistema consistirá en generar eventos y verificar que son detectados, además del tiempo de respuesta entre que ocurre la acción y es mostrada en el Grafana.

3.1.1 Escenarios de pruebas.

Se definieron los siguientes escenarios de pruebas:

- ***Métricas de los servidores.***
  - Servidor Proxmox8 (Sistema operativo Proxmox 4.2.23)
  - PC015381 (LinuxMint17)
- ***Trazas de Squid Proxy.***
  - Período 24052018-24052018
- ***Eventos asociados a la autenticación de usuario que fueron definidos como requerimientos.***

- PC015381 (LinuxMint17)
- Servidor virtual LOGelastic (Ubuntu16.04)

3.2 Eventos generados y pruebas con Logs de Squid.

Métricas de los servidores.

Se verifica localmente los valores de métricas asociadas a: CPU, HDD y RAM de los equipos seleccionados para comparar con lo mostrado en el Dashboard de Grafana. Posteriormente se bajarán los umbrales de las alarmas para intencionar su reporte.

- Servidor Proxmox8 (Sistema operativo Proxmox 4.2.23)

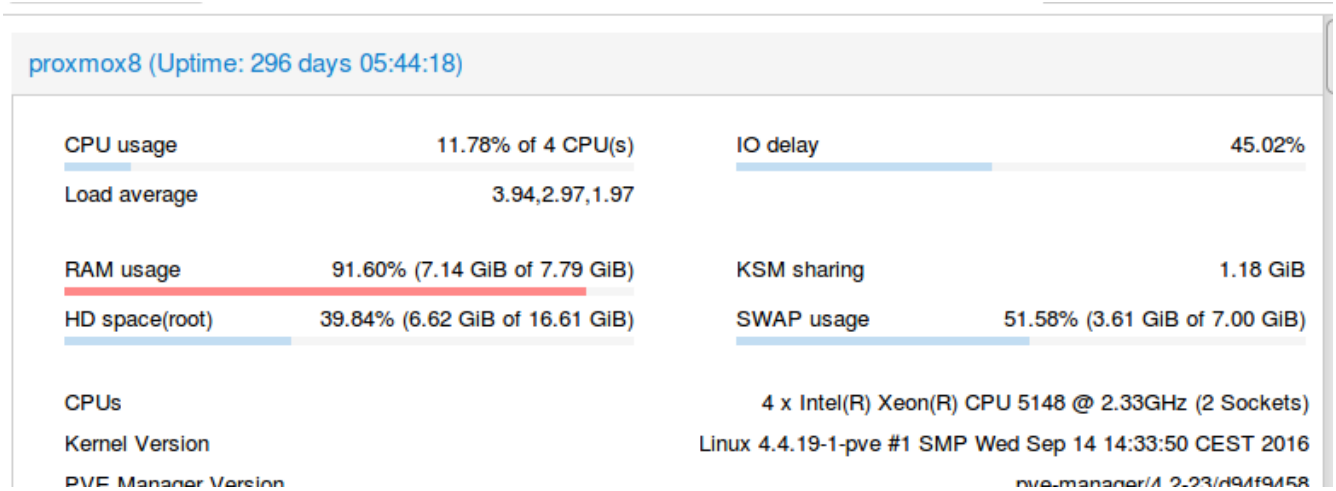

Fig.19 Uso de CPU, memoria RAM, DISK.

- PC015381 (Sistema Operativo Linux Mint 17).

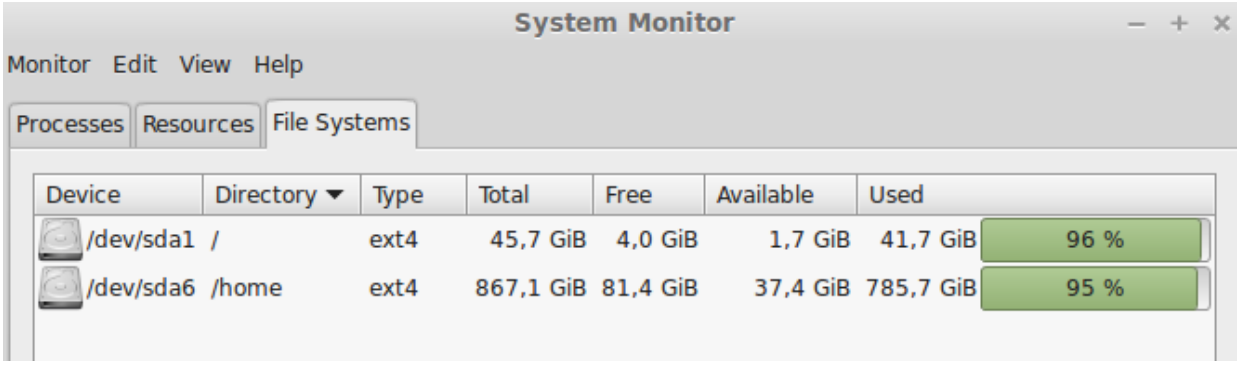

Fig. 20 Uso de almacenamiento en PC15381.

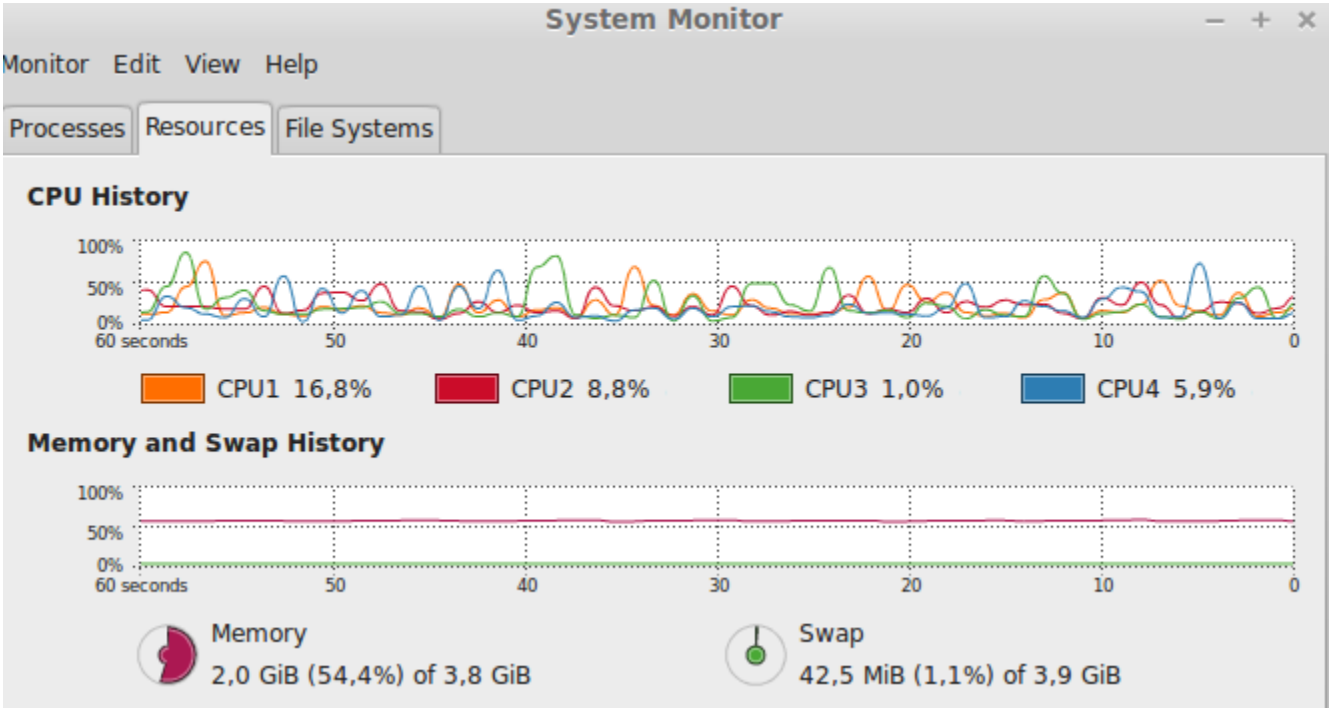

Fig. 21 Uso de CPU y RAM de PC15381.

Trazas de Squid Proxy.

Se analiza el período 24052018-24052018 con la herramienta SARG para comparar los resultados de top de usuarios y el detallado de tres de ellos. Fig.21

Sarg Squid Analysis Report Generator

Squid User Access Reports

Period: 2018 may 24—2018 may 25
Sort: bytes, reverse
Top users

Top sites
Sites & Users
Downloads
Denied accesses
Authentication Failures

NUM		USERID	CONNECT	BYTES	%BYTES	IN-CACHE-OUT		ELAPSED TIME	MILLISEC	%TIME
1	G T	fredy.delsol	90	9.76G	20,15%	100,00%	0,00%	25:05:45	90,345,297	1,47%
2	G T	ramon.lopez	6.11K	4.90G	10,12%	99,57%	0,43%	90:24:18	325,458,952	5,31%
3	G T	niuris.mendez	805	3.86G	7,98%	99,98%	0,02%	33:05:52	119,152,288	1,94%
4	G T	miguel.borrell	1.00K	3.33G	6,88%	99,69%	0,31%	39:15:42	141,342,602	2,31%
5	G T	jose.suri	968	2.56G	5,30%	99,86%	0,14%	48:01:23	172,883,185	2,82%

Fig.21 Herramienta Sarg.

Eventos asociados a la autenticación de usuario que fueron definidos como requerimientos.

- PC015381 (LinuxMint17)
  - Conexión por ssh fallida por clave.
  - Múltiples intentos de conexión por ssh fallida por usuario, usuario y clave.
  - Conexión exitosa por ssh.
  - Acceso como root.
  - Intento fallido de escalar privilegios.
- Servidor virtual KASPERSKY-12R2 (Windows2012R2)
  - Crear/Eliminar usuario.
  - Asignar privilegios administrativos a usuario.

- Login como administrador.
- Escalar privilegios.

- Servidor virtual LOGelastic (Ubuntu16.04)
  - Conexión por ssh fallida por clave.
  - Múltiples intentos de conexión por ssh fallida por usuario, usuario y clave.
  - Conexión exitosa por ssh.
  - Acceso como root.
  - Intento fallido de escalar privilegios.

3.3 Resultados obtenidos.

3.3.1 Métricas de los servidores.

Se corresponden los valores locales con los mostrados por Grafana.

Servidor Proxmox8 (Sistema operativo Proxmox 4.2.23)

Datos mostrados en la herramienta Grafana en cuanto uso de : CPU, RAM y DISK. Fig. 22

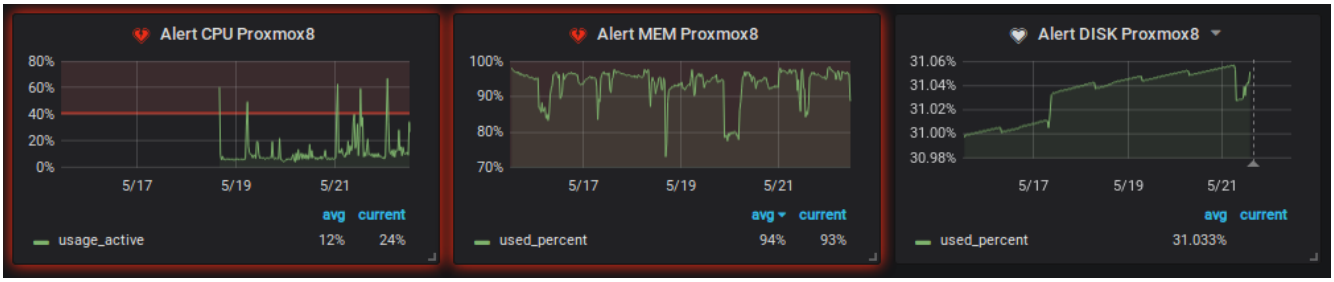

Fig.22 Uso de CPU, RAM y DISK en el Servidor Proxmox8 mostrado en Grafana.

PC015381 (LinuxMint17)

Se muestra el uso de CPU, RAM y DISK los cuales corresponden con los datos locales de la PC015381. Fig.23

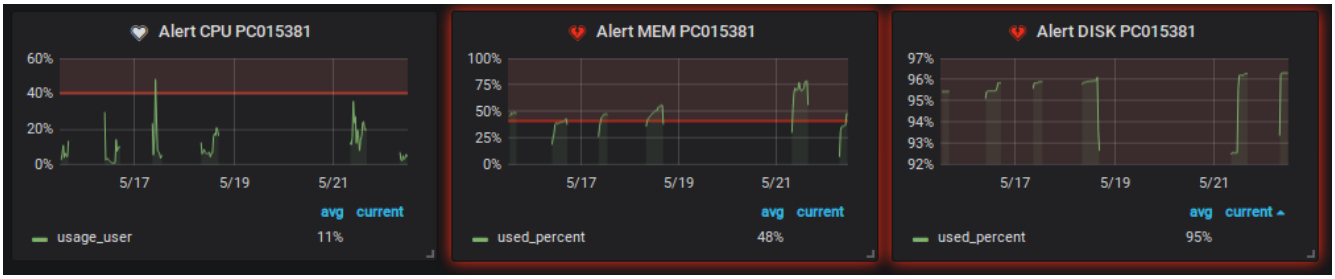

Fig.23 Uso de CPU, RAM y DISK en PC015381 mostrado en Grafana.

3.3.1.1 Alarmas de los servidores.

Se analizan las alarmas generadas por los equipos seleccionados asociadas a: CPU, HDD y RAM. Para esto se bajara el umbral de alerta a un 40%.

Servidor Proxmox8 (Sistema operativo Proxmox 4.2.23) Fig:24

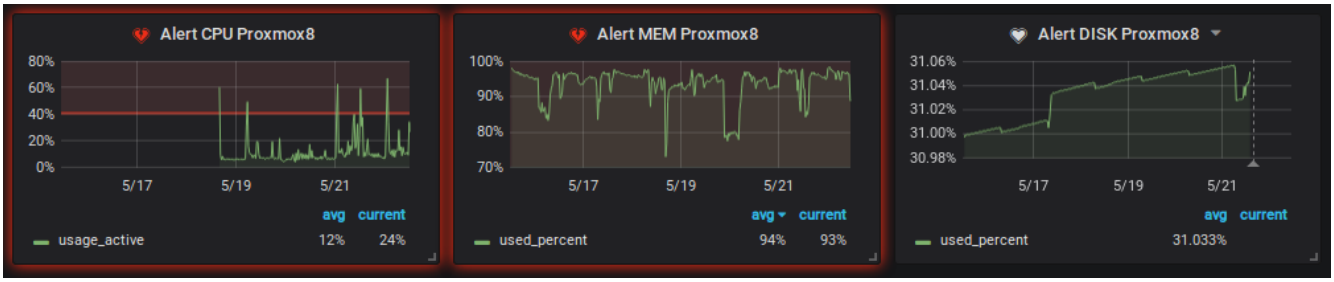

Fig.24 Alarmas en el Servidor Proxmox8.

Se aprecia en la Fig.24 como la CPU y la MEM del servidor Proxmox 8 estan por encima del 40% de su capacidad, generando su debida alerta.

PC015381 (LinuxMint17) Fig.25

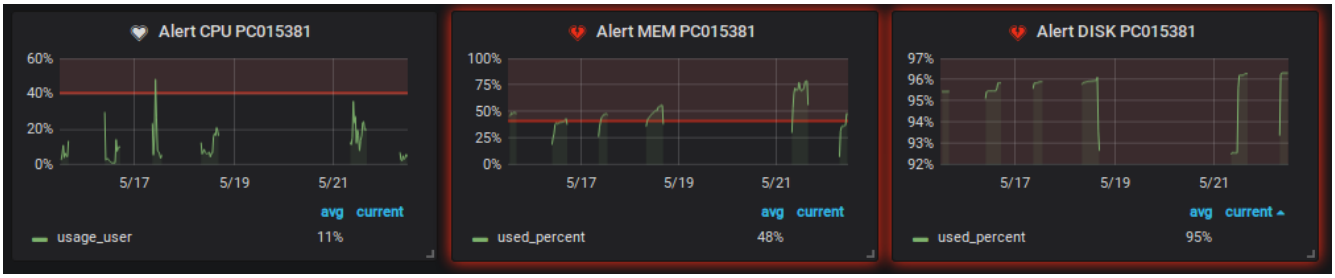

Fig.24 Alarmas en PC015381.

Se observa la activacion de la alerta cunado existe un uso mayor del 40% de la alarma relacionada con la MEM y DISK.

Cuando una alerta cambia de estado, envia notificaciones. Cada regla de alerta puede tener múltiples notificaciones, estan se muestran en el mismo Grafana y en nuestro caso se configuró un canal de notificaciones las cuales llegarán por correo electrónico.

Aviso de alerta en la herramienta Grafana. Fig.25.

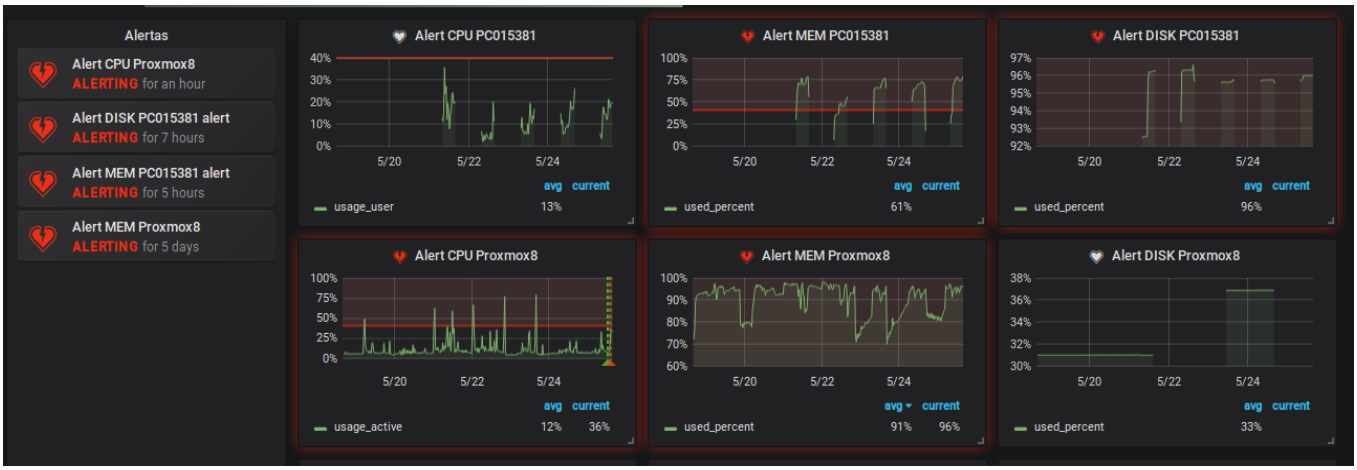

Fig.25 Aviso de las alertas mostrados en Grafana.

Aviso por correo electrónico. Fig.26 .

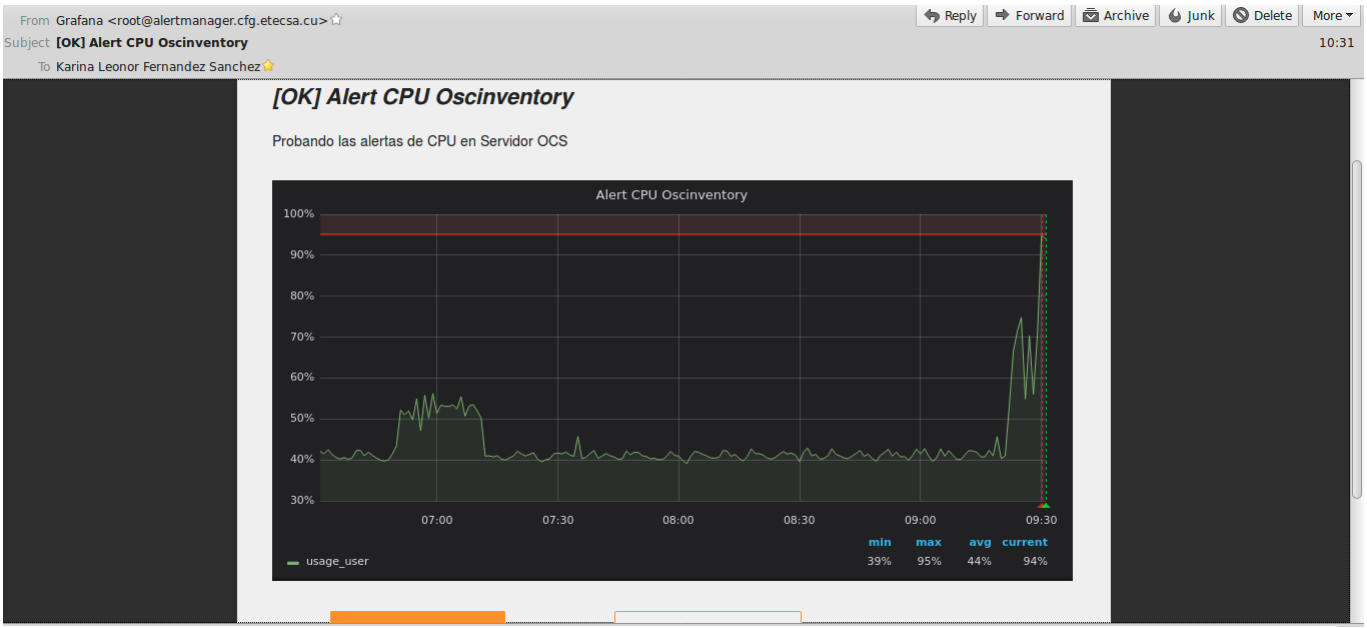

Fig.26 Aviso de alerta enviado por correo electrónico.

3.3.2 Trazas de Squid Proxy.

Se muestran en las trazas de Squid Proxy el tráfico usado por cada usuario y el sitio en donde mas visitas realizó.

Para corroborar que la información mostrada es correcta, se comparan tres usuarios dados por la herramienta SARG y se verifican en Grafana. Fig.27,28,29.

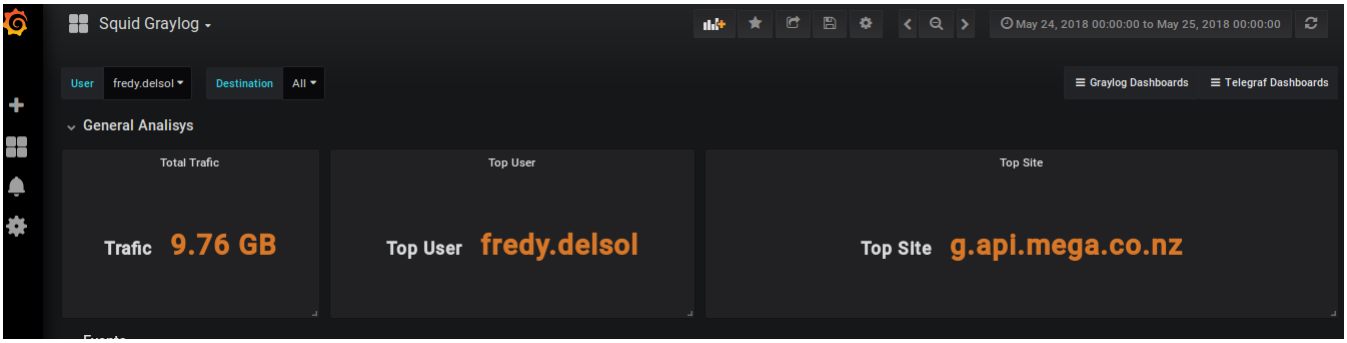

Fig.27 Tráfico del usuario Fredy del Sol y el sitio al que más accedió.

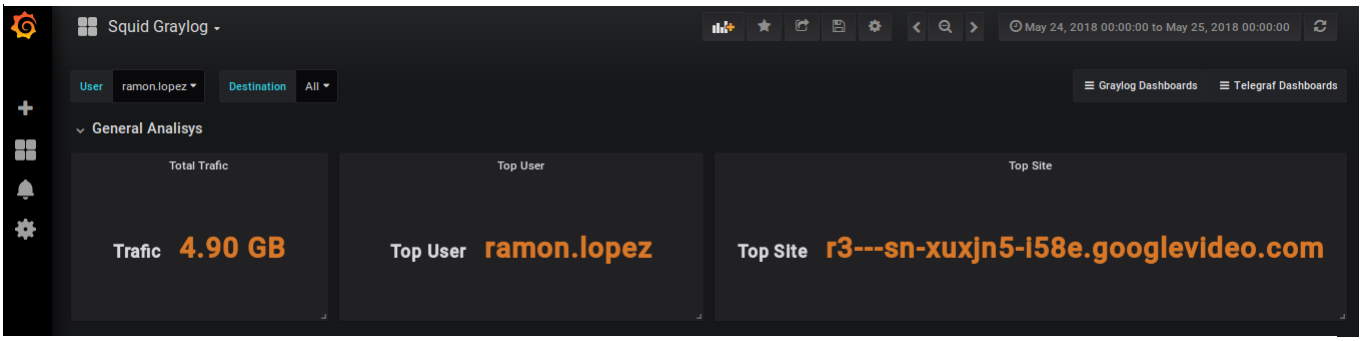

Fig.28 Tráfico del usuario Ramon Lopez y el sitio al que más accedió.

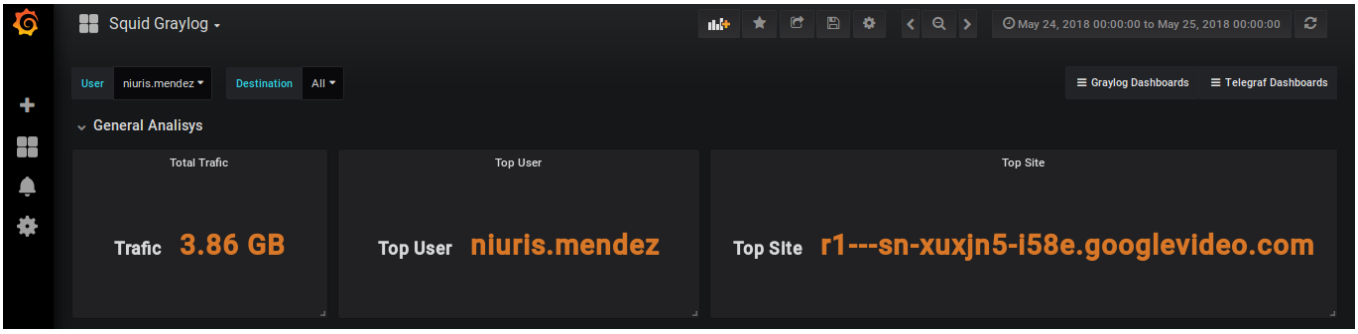

Fig.29 Tráfico del usuario Niuris Mendez y el sitio al que más accedió.

3.3.3 Eventos de privilegios administrativos.

En la Fig.30 se muestran los registros de eventos cuando un usuario intenta una conexión por ssh y esta es fallida por clave o nombre de usuario o cuando esta es exitosa.

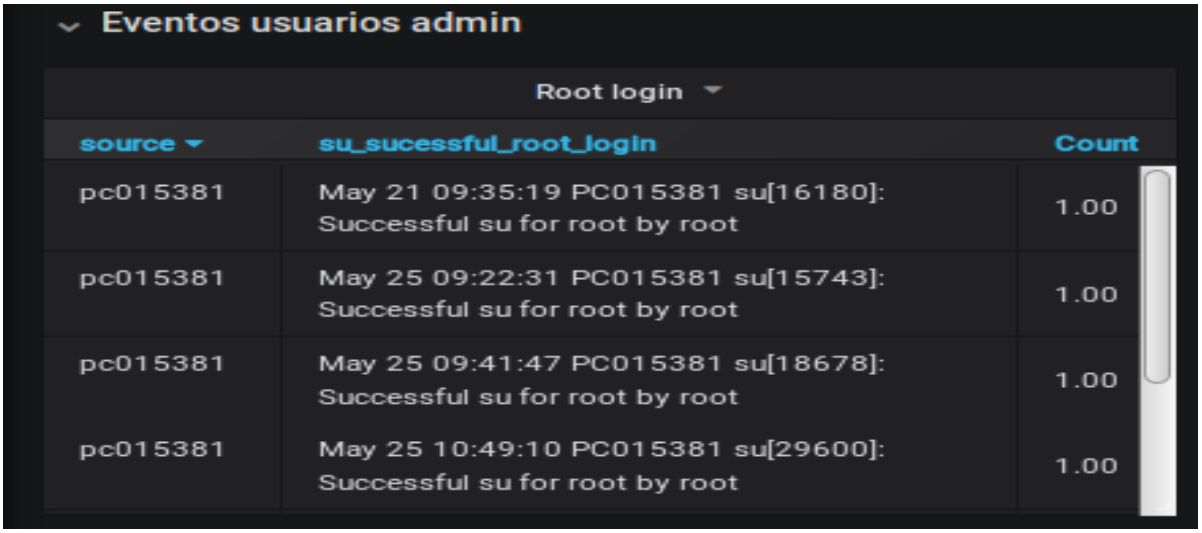

Fig.30 Eventos de SSH.

En la fig.31 se muestra como queda registrado cuando un usuario realiza un acceso root (sudo su).

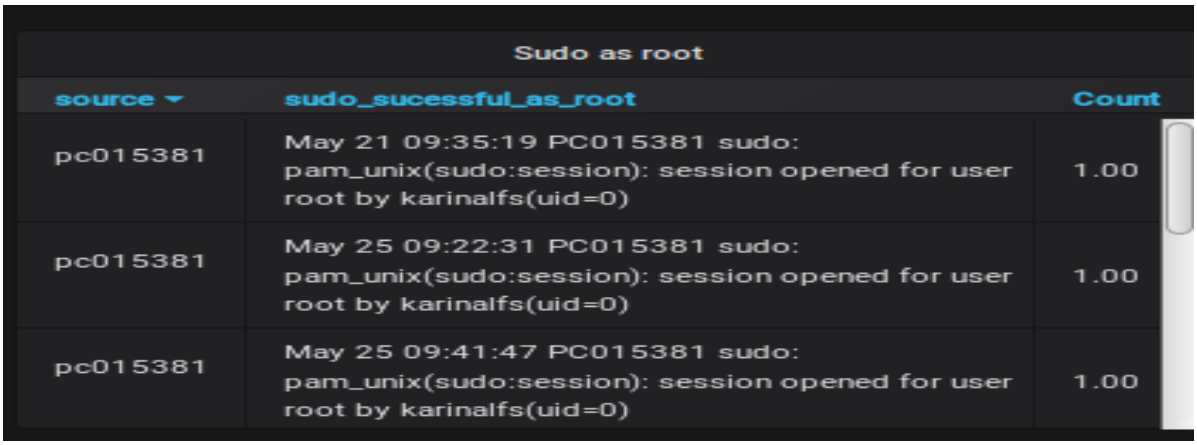

Fig.31 Usuario realiza un acceso root (sudo su).

Al realizar un intento fallido de escala privilegios este quedará registrado en el panel de Sudo

failure. Fig.32.

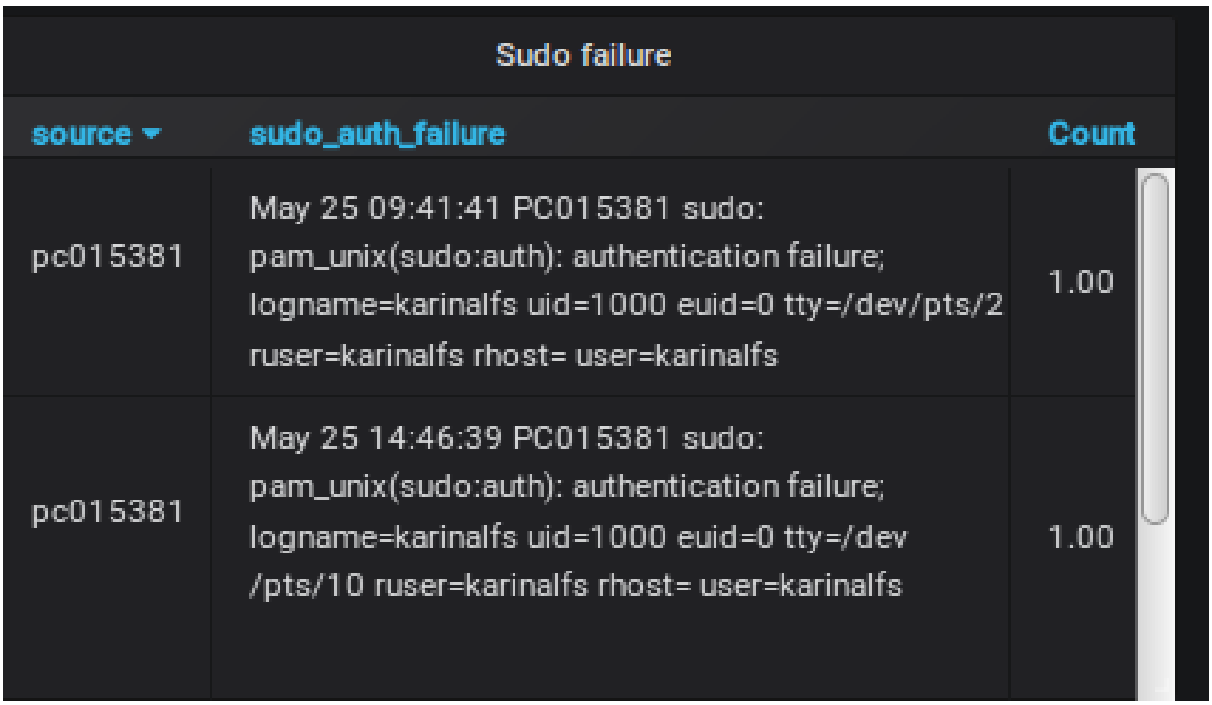

Fig.32 Intento fallido de escala privilegios.

3.3.3.1 SSH

Se muestran los eventos de ssh generdos como: logeo SSh satifactorio, fallo de logueo, ataque de fuerza bruta por SSH. Fig.33

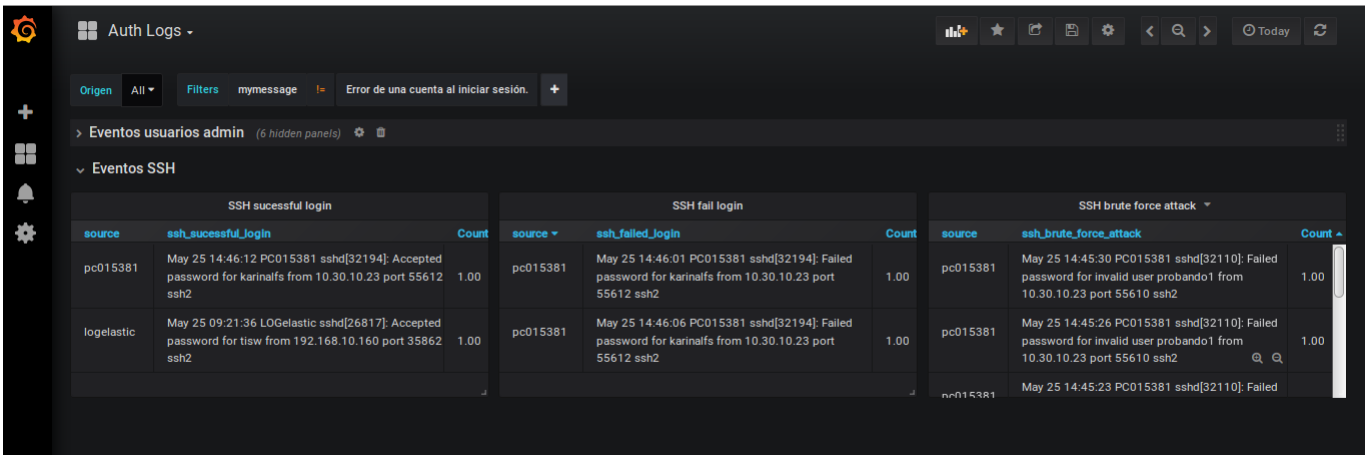

Fig.33: Logeo SSh satifactorio, fallo de logueo, ataque de fuerza bruta por SSH.

3.3.3.2 Servidor virtual Kaspersky-12R2 (Windows2012R2).

Se mueastra cuando en un servidor se crea o se elimina un usuario, se asigne privilegios administrativos a usuario, un logueo como administrador y escala de privilegios. Fig.34

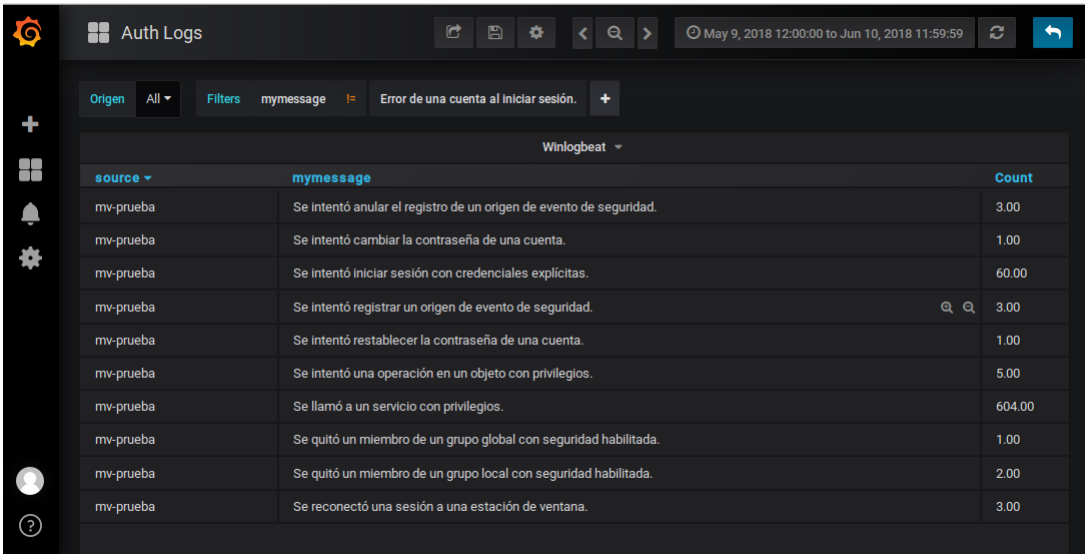

Fig.34 Eventos relacionados con el Servidor virtual Kaspersky-12R2

3.3.3.3 Servidor virtual LOGelastic (Ubuntu 16.04)

En la Fig.35,36 se muestra el registro de eventos asociados a la autenticación de usuarios generados por el servidor virtual LOGelastic tales como el intento de realizar una conexión por ssh y esta es fallida por clave o usuario. La conexión ha sido exitosa, acceso como root o intentos fallidos al escalar privilegios.

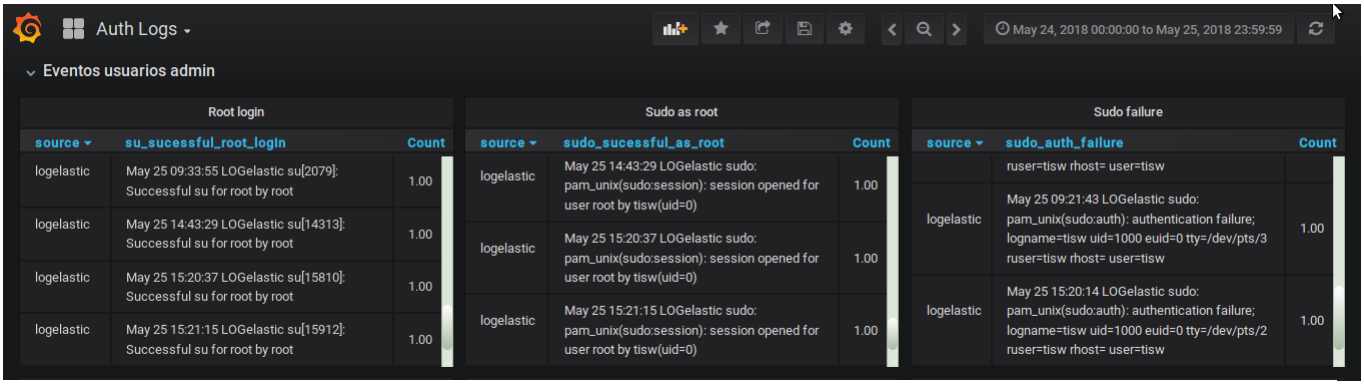

Fig.35 Eventos relacionados con la autenticación de usuario generados por el servidor virtual LOGelastic.

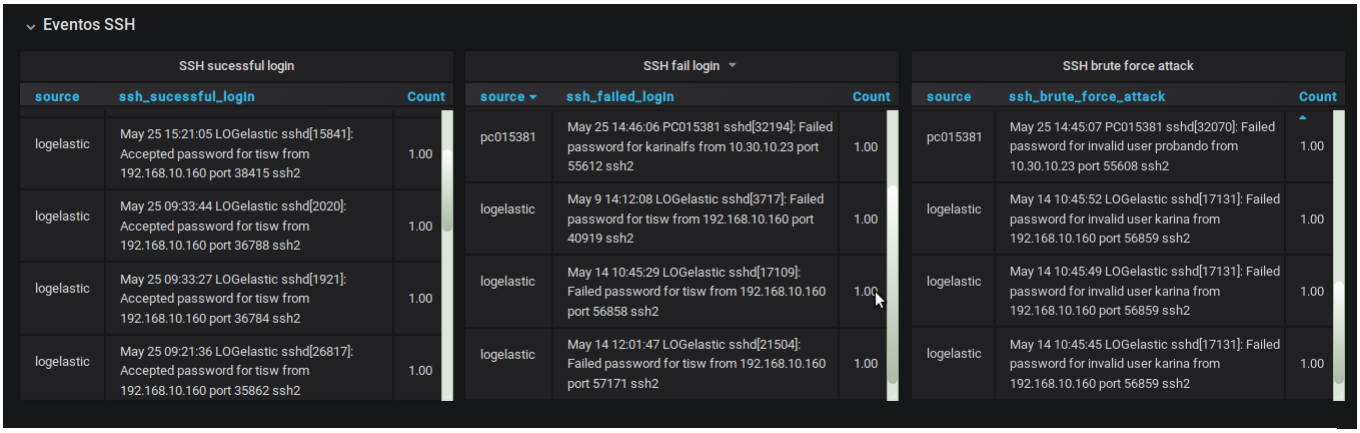

Fig.36 Eventos genereados por SSH por el servidor virtual LOGelastic.

3.4 Conclusiones parciales.

- Se comprueba que lo mostrado en la herramienta Grafana correspone con lo que esta realmente sucediendo en los servidores, equipos y sistemas operativos.
-

Conclusiones:

1. Existe gran diversidad de aplicaciones que son utilizadas para la gestión de logs. Para satisfacer los requerimientos solicitados por la Dirección Territorial de ETECSA en Cienfuegos en cuanto a la Gestión de Logs es necesario implementar varias herramientas open source. Se seleccionan como clientes a implementar: filebeat, winlogbeat y telegraf. Graylog es seleccionado como colector de logs y se encarga de su envío indexado hasta Elasticsearch. Las métricas de los servidores serán almacenadas en InfluxDB. Grafana será el visor utilizado para graficar los eventos de sistema. Por el consumo de recursos Graylog, Elasticsearch y Grafana están instalados en servidores independientes.

2. Se realizó un Diseño de Sistema de Gestión de Logs basado en software libre que responde a las necesidades de la Dirección Territotial de ETECSA en Cienfuegos. Las herramientas seleccionadas permiten continuar incorporando nuevos elementos de monitoreo, lo que permite seguir desarrollando el sistema de gestión.

3. Se comprueba que lo mostrado en la herramienta Grafana correspone con lo que esta realmente sucediendo en los servidores, equipos y sistemas operativos, a través de eventos generados intencionalmente para poder corroborar la información mostrada.

Recomendaciones:

Implementar el sistema propuesto en las demás Divisiones Territoriales de ETECSA del país.

Referencias Bibliográficas:

1 .A. D. Castro, «Comenzando con Elasticsearch», *Byteflair*, 17-jul-2013.
2. «El mercado de las herramientas de monitorización», 07-sep-2017. .
3. «Extended Log Format», *Wikipedia, la enciclopedia libre*. 22-may-2017.
4. Colectivo de autores, «Gestión de Bitácoras (Logs) de Eventos», 06-nov-2017.
5. L. F. D. Vega, «Monitor de Squid con InfluxDB y Grafana», *Sysadmins de Cuba*. [En línea]. Disponible en: https://www.sysadminsdecuba.com/2018/03/monitorizando-las-conexiones-de-squid-con-influxdb-y-grafana/. [Accedido: 15-may-2018].
6. Alexander Rivas, «Parseando Logs de Squid con Graylog y Mostrándolos en Grafana. 4ta. Parte», *El Blog de Jorge de la Cruz*, 01-may-2018. [En línea]. Disponible en: https://www.jorgedelacruz.es/2018/05/01/parseando-logs-de-squid-con-graylog-y-mostrandolos-en-grafana-4ta-parte/. [Accedido: 15-may-2018].
7. «proxmox8 - Proxmox Virtual Environment». [En línea]. Disponible en: https://192.168.80.8:8006/#v1:0:=qemu%2F153:4:::::: [Accedido: 09-nov-2016].
8. «Send logs from Windows without agent / winlogbeat / filebeat», *Discuss the Elastic Stack*. [En línea]. Disponible en: https://discuss.elastic.co/t/send-logs-from-windows-without-agent-winlogbeat-filebeat/120600. [Accedido: 14-may-2018].
9. «Top 51 Log Management Tools for Monitoring, Analytics and more». [En línea]. Disponible en: https://stackify.com/best-log-management-tools/. [Accedido: 14-may-2018].
10. «top de herramientas para gestion de logs - Buscar con Google». [En línea]. Disponible en: https://www.google.com.cu/search?source=hp&ei=WvX6WqaqC42RjwTMhr3ICg&q=top+de+herramientas+para+gestion+de+logs&oq=top+de+herramientas+para+gestion+de+logs&gs_l=psy-ab.3...1227.12604.0.13267.47.42.1.0.0.0.373.6418.0j21j8j3.32.0....0...1c.1.64.psy-ab..14.31.6041.0..0j35i39k1j0i131k1j0i22i30k1j0i8i13i30k1j0i22i10i30k1j33i22i29i30k1.0.ZQQDZeBtR5Q. [Accedido: 15-may-2018].
11. «Using both Filebeat and Winlogbeat - Beats / Filebeat - Discuss the Elastic Stack». [En línea]. Disponible en: https://discuss.elastic.co/t/using-both-filebeat-and-winlogbeat/103484. [Accedido: 14-may-2018].

Bibliogrfía:

[1] «La importancia de un SGSI». [En línea]. Disponible en: https://www.welivesecurity.com/la-es/2010/09/10/la-importancia-de-un-sgsi/. [Accedido: 30-nov-2017].
[2] «La importancia de tener un buen log». [En línea]. Disponible en: http://software-tecnico-libre.es/es/articulo-por-tema/todas-las-secciones/todos-los-temas/todos-los-articulos/importancia-del-log. [Accedido: 15-may-2018].
[3] «La importancia de los registros de sistema - Security Art Work». [En línea]. Disponible en: https://www.securityartwork.es/2010/03/22/la-importancia-de-los-registros-de-sistema/. [Accedido: 30-nov-2017].
[4] «¿Qué son los Logs y por qué deben interesarte?» [En línea]. Disponible en: http://dbi.io/es/blog/que-son-los-logs/. [Accedido: 30-nov-2017].
[5] «Monitoreo de Redes: Las 16 mejores herramientas de monitorizacion de redes de 2016». [En línea]. Disponible en: https://blog.pandorafms.org/es/herramientas-de-monitoreo-de-redes/. [Accedido: 04-dic-2017].
[6] «Caso de uso: ¿Cómo y por qué centralizar tus logs?», *DBI*, 06-abr-2017. [En línea]. Disponible en: http://dbi.io/es/blog/caso-uso-centralizar-logs/. [Accedido: 15-may-2018].
[7] seo, «Herramientas para la Monitorización de redes de 2018», *Axarnet*, 03-abr-2018. [En línea]. Disponible en: https://www.axarnet.es/blog/herramientas-monitorizacion-redes/. [Accedido: 15-may-2018].
[8] «Las 10 mejores herramientas de monitoreo de redes del 2017 | Apen: Soluciones informáticas». [En línea]. Disponible en: https://apen.es/2017/03/10/las-10-mejores-herramientas-de-monitoreo-de-redes-del-2017/. [Accedido: 15-may-2018].
[9] «Extended Log Format», *Wikipedia, la enciclopedia libre*. 22-may-2017.
[10] «Windows Event Log Structures (Windows)». [En línea]. Disponible en: https://msdn.microsoft.com/en-us/library/windows/desktop/aa385786(v=vs.85).aspx. [Accedido: 15-may-2018].
[11] «Registro de eventos de seguridad del sistema Windows». [En línea]. Disponible en: https://www.quest.com/mx-es/products/intrust/. [Accedido: 15-may-2018].
[12] «Control de logs en sistemas linux - logsLinux.pdf». [En línea]. Disponible en: https://hardsoftsecurity.es/logsLinux.pdf. [Accedido: 15-may-2018].
[13] «Logs en Linux | Estréllate y Arde.ORG». [En línea]. Disponible en: http://www.estrellateyarde.org/logs-en-linux. [Accedido: 15-may-2018].
[14] «Monitorizacion de Logs: ¿qué debemos hacer antes de comenzar?» [En línea]. Disponible en: https://blog.pandorafms.org/es/monitorizacion-de-logs/. [Accedido: 15-may-2018].
[15] «100+ Top Server Monitoring & Application Monitoring Tools», *Linux Systems Analyst | Hayden James*, 07-mar-2018. [En línea]. Disponible en: https://haydenjames.io/50-top-server-monitoring-application-performance-monitoring-apm-solutions/. [Accedido: 15-may-2018].
[16] «150 Open Source, Free and Top Dashboard Software in 2018». [En línea]. Disponible en: https://www.predictiveanalyticstoday.com/open-source-dashboard-software/. [Accedido: 15-may-2018].
[17] «4 Best Event Log Analysis Tools & Software for Windows/Open Source (FREE & PAID) • ITT Systems», *ITT Systems*. [En línea]. Disponible en: https://www.ittsystems.com/best-event-log-analysis-tools/. [Accedido: 14-may-2018].

[18] «Using both Filebeat and Winlogbeat», *Discuss the Elastic Stack*. [En línea]. Disponible en: https://discuss.elastic.co/t/using-both-filebeat-and-winlogbeat/103484/4. [Accedido: 15-may-2018].
[19] «Monitoring Filebeat | Filebeat Reference [6.2] | Elastic». [En línea]. Disponible en: https://www.elastic.co/guide/en/beats/filebeat/current/monitoring.html. [Accedido: 14-may-2018].
[20] «Configure the output | Filebeat Reference [6.2] | Elastic». [En línea]. Disponible en: https://www.elastic.co/guide/en/beats/filebeat/current/configuring-output.html. [Accedido: 14-may-2018].
[21] «Winlogbeat Reference [6.2] | Elastic». [En línea]. Disponible en: https://www.elastic.co/guide/en/beats/winlogbeat/current/index.html. [Accedido: 14-may-2018].
[22] «Telegraf from InfluxData | Agent for Collecting & Reporting Metrics & Data». [En línea]. Disponible en: https://www.influxdata.com/time-series-platform/telegraf/. [Accedido: 14-may-2018].
[23] «Logstash». [En línea]. Disponible en: https://www.elastic.co/products/logstash. [Accedido: 14-may-2018].
[24] «Graylog | Open Source Log Management». [En línea]. Disponible en: https://www.graylog.org/. [Accedido: 14-may-2018].
[25] «Graylog 2 VS PandoraFMS comparación en profundidad». [En línea]. Disponible en: https://blog.pandorafms.org/es/graylog-2/. [Accedido: 14-may-2018].
[26] «Elasticsearch - EcuRed». [En línea]. Disponible en: https://www.ecured.cu/Elasticsearch. [Accedido: 15-may-2018].
[27] «¿Qué es InfluxDB y como funciona?» [En línea]. Disponible en: https://www.okitup.com/blog/que-es-influxdb/. [Accedido: 15-may-2018].
[28] «Creando un sistema de visualización de series temporales IV: Conceptos básicos de InfluxDB – Sitio Web de Roberto Orayen». [En línea]. Disponible en: https://robertoorayen.eu/2016/11/21/conceptos-basicos-influxdb/. [Accedido: 15-may-2018].
[29] «Introducción a Kibana | adictosaltrabajo». [En línea]. Disponible en: https://www.adictosaltrabajo.com/tutoriales/introduccion-a-kibana/. [Accedido: 15-may-2018].
[30] «Grafana, ¿Cómo restringir el acceso a las organizaciones usando un controlador de dominio? – Código 200». [En línea]. Disponible en: https://codigo200.wordpress.com/2017/11/21/grafana-como-restringir-el-acceso-a-las-organizaciones-usando-un-controlador-de-dominio/. [Accedido: 15-may-2018].
[31] «Grafana — documentación de KEVELL - 0.1». [En línea]. Disponible en: http://kevellspanish.readthedocs.io/es/latest/sgrafana_deploy.html. [Accedido: 15-may-2018].
[32] R. Páez, «Análisis forense en sistemas Linux – Obteniendo información (Parte 2)», *Security Art Work*, 30-may-2012. [En línea]. Disponible en: https://www.securityartwork.es/2012/05/30/analisis-forense-en-sistemas-linux-obteniendo-informacion-parte-2/. [Accedido: 16-may-2018].
[33] Omar, «Clúster Elasticsearch. Logstash y Grafana. Análisis de logs. 3ra Parte.», *Sysadmins de Cuba*. [En línea]. Disponible en: https://www.sysadminsdecuba.com/2017/12/cluster-elasticsearch-logstash-y-grafana-analisis-de-logs-3ra-parte/. [Accedido: 15-may-2018].
[34] Alexander Rivas, «Cómo instalar y configurar Graylog en Ubuntu 16.04. 1era Parte.», *El Blog de Jorge de la Cruz*, 10-abr-2018. [En línea]. Disponible en: https://www.jorgedelacruz.es/2018/04/10/como-instalar-y-configurar-graylog-en-ubuntu-16-04-1era-parte/. [Accedido: 14-may-2018].

[35] Alexander Rivas, «Parseando Logs de Zimbra con Graylog y Mostrándolos en Grafana. 2da Parte.», *El Blog de Jorge de la Cruz*, 11-abr-2018. [En línea]. Disponible en: https://www.jorgedelacruz.es/2018/04/11/parseando-logs-de-zimbra-con-graylog-y-mostrandolos-en-grafana-2da-parte/. [Accedido: 17-may-2018].
[36] «El SerDe de Grok - Amazon Athena». [En línea]. Disponible en: https://docs.aws.amazon.com/es_es/athena/latest/ug/grok.html. [Accedido: 18-may-2018].
[37] «Extractors — Graylog 2.4.4 documentation». [En línea]. Disponible en: http://docs.graylog.org/en/2.4/pages/extractors.html. [Accedido: 18-may-2018].
[38] «En busca del Dashboard perfecto: InfluxDB, Telegraf y Grafana - Parte I», *El Blog de Jorge de la Cruz*, 29-jun-2016. [En línea]. Disponible en: https://www.jorgedelacruz.es/2016/06/29/en-busca-del-dashboard-perfecto-influxdb-telegraf-y-grafana-parte/. [Accedido: 15-may-2018].
[39] «telegraf», *Docker Documentation*, 17-may-2018. [En línea]. Disponible en: https://docs.docker.com/samples/library/telegraf/. [Accedido: 18-may-2018].

Glosario de términos:

Machine Learning: disciplina científica del ámbito de la Inteligencia Artificial que crea sistemas que aprenden automáticamente.

SLA: Service Level Agreement (SLA) es un contrato que describe el nivel de servicio que un cliente espera de su proveedor. En español, también se llama Acuerdo de Nivel de Servicio (ANS).

Syslog: Es un estándar para el registro de mensajes . Permite la separación del software que genera los mensajes, el sistema que los almacena y el software que los informa y analiza. Cada mensaje está etiquetado con un código de instalación, que indica el tipo de software que genera el mensaje, y se le asigna una etiqueta de gravedad.

Squid: Es un popular programa de software libre que implementa un servidor proxy y un dominio para caché de páginas web, publicado bajo licencia GPL(General Public Licence). Tiene una amplia variedad de utilidades, desde acelerar un servidor web, guardando en caché peticiones repetidas a DNS(Servidor de nombre de dominio) y otras búsquedas para un grupo de gente que comparte recursos de la red, hasta caché de Web, además de añadir seguridad filtrando el tráfico.

Visor de eventos: Permite el monitoreo y manejo del sistema, Seguridad, y los eventos de aplicación y los errores del sistema.

Daemon: Es un tipo especial de proceso informático no interactivo, es decir, que se ejecuta en segundo plano en vez de ser controlado directamente por el usuario.

SSH o Secure Shell: Es un protocolo de administración remota que permite a los usuarios controlar y modificar sus servidores remotos a través de Internet.

Kernel: Software que constituye una parte fundamental del sistema operativo, y se define como la parte que se ejecuta en modo privilegiado (conocido también como modo núcleo). Es el principal responsable de facilitar a los distintos programas acceso seguro al hardware de la computadora o en forma básica, es el encargado de gestionar recursos, a través de servicios de llamada al sistema.

RPM: Red Hat Package Manager, es una herramienta de administración de paquetes pensada básicamente para GNU/Linux. Es capaz de instalar, actualizar, desinstalar, verificar y solicitar programas. RPM es el formato de paquete de partida del Linux Standard Base.

Yum: Es un gestor de paquetes desarrollado para mejorar la instalación de los RPMs.

Sudo: Super user do es una utilidad de los sistemas operativos tipo Unix, como Linux, BSD, o Mac OS X, que permite a los usuarios ejecutar programas con los privilegios de seguridad de otro usuario (normalmente el usuario root) de manera segura, convirtiéndose así temporalmente en superusuario. Se instala por defecto en /usr/bin.

LDAP: Lightweight Directory Access Protocol (en español Protocolo Ligero/Simplificado de Acceso a Directorios) que hacen referencia a un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. LDAP también se considera una base de datos (aunque su sistema de almacenamiento puede ser diferente) a la que pueden realizarse consultas.

Active Directory: Término que utiliza Microsoft para referirse a su implementación de servicio de directorio en una red distribuida de computadores. Utiliza distintos protocolos, principalmente LDAP, DNS, DHCP y Kerberos.

JSON: Acrónimo de JavaScript Object Notation, es un formato de texto ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript aunque hoy, debido a su amplia adopción como alternativa a XML, se considera un formato

de lenguaje independiente.

TI: Técnologia de la informática

Curl: Herramienta de línea de comandos y biblioteca para transferir datos con URL

JOIN: (unir, combinar) de SQL permite combinar registros de una o más tablas en una base de datos relacional. En el Lenguaje de Consultas Estructurado (SQL)

Timestamp: Una marca temporal, conocida también como registro de tiempo, es una secuencia de caracteres que denotan la hora y fecha (o alguna de ellas) en la/s que ocurrió determinado evento.

DMZ: Es una zona segura que se ubica entre la red interna de una organización y una red externa, generalmente en Internet. El objetivo de una DMZ es que las conexiones desde la red interna y la externa a la DMZ estén permitidas, mientras que en general las conexiones desde la DMZ solo se permitan a la red externa -- los equipos (hosts) en la DMZno pueden conectar con la red interna. Esto permite que los equipos (hosts) de la DMZ puedan dar servicios a la red externa a la vez que protegen la red interna en el caso de que intrusos comprometan la seguridad de los equipos (host) situados en la zona desmilitarizada. Para cualquiera de la red externa que quiera conectarse ilegalmente a la red interna, la zona desmilitarizada se convierte en un callejón sin salida.

Anexos:

Anexo 1

Versiones compatibles de productos Elastic.

Elasticsearch	Kibana	X-Pack	Beats*	Logstash	ES-Hadoop (jar)
5.0.x	5.0.x	5.0.x	1.3.x-5.6.x	2.4.x-5.6.x	5.0.x-5.6.x
5.1.x	5.1.x	5.1.x	1.3.x-5.6.x	2.4.x-5.6.x	5.0.x-5.6.x
5.2.x	5.2.x	5.2.x	1.3.x-5.6.x	2.4.x-5.6.x	5.0.x-5.6.x
5.3.x	5.3.x	5.3.x	1.3.x-5.6.x	2.4.x-5.6.x	5.0.x-5.6.x
5.4.x	5.4.x	5.4.x	1.3.x-5.6.x	2.4.x-5.6.x	5.0.x-5.6.x
5.5.x	5.5.x	5.5.x	1.3.x-5.6.x	2.4.x-5.6.x	5.0.x-5.6.x
5.6.x	5.6.x	5.6.x	1.3.x-6.0.x	2.4.x-6.0.x	5.0.x-6.0.x
6.0.x	6.0.x	6.0.x	5.6.x-6.2.x	5.6.x-6.2.x	6.0.x-6.2.x
6.1.x	6.1.x	6.1.x	5.6.x-6.2.x	5.6.x-6.2.x	6.0.x-6.2.x
6.2.x	6.2.x	6.2.x	5.6.x-6.2.x	5.6.x-6.2.x	6.0.x-6.2.x

Esta información está disponible en:

https://www.elastic.co/support/matrix#matrix_compatibility

Anexo 2

Content Pack para Squid-Graylog

{

"name":"Squid Content Pack",

"description":"Grok Patterns, Inputs, extractors and stram.",

"category":"Proxies",

"inputs":[

Anexos


```
{
"id":"5a90a5a0687cf821d00466f4",
"title":"Squid-Logs",
"configuration":{
"recv_buffer_size":1048576,
"port":5044,
"tls_key_file":"",
"tls_enable":false,
"tls_key_password":"",
"tcp_keepalive":false,
"tls_client_auth_cert_file":"",
"tls_client_auth":"disabled",
"override_source":null,
"bind_address":"0.0.0.0",
"tls_cert_file":""
},
"static_fields":{

},
"type":"org.graylog.plugins.beats.BeatsInput",
"global":false,
```

"extractors":[

{

"title":"Squid Extractor",

"type":"GROK",

"cursor_strategy":"COPY",

"target_field":"",

"source_field":"message",

"configuration":{

"grok_pattern":"^%{BASE10NUM:epoch_time}%{SPACE:UNWANTED}%{INT:Duration;int} %{IPV4:Client_Address} %{DATA:Result_Code;string}\\/%{BASE10NUM:Status_Code;int} %{INT:Bytes;int} %{WORD:Method;string} %{SQUIDURI:URI;string} %{DATA:User;string} %{DATA:Peer_Status;string}\\/%{DATA:Peer_Host;string} %{DATA:Content-Type;string}$",

"named_captures_only":false

},

"converters":[

],

"condition_type":"NONE",

"condition_value":"",

"order":0

},

{

"title":"Unix Time Extractor",

"type":"SUBSTRING",

"cursor_strategy":"COPY",

"target_field":"epoch_time",

"source_field":"epoch_time",

"configuration":{

"end_index":10,

"begin_index":0

},

"converters":[

],

"condition_type":"NONE",

"condition_value":"",

"order":0

}

]

}

],

"streams":[

{

"id":"5a9479ed687cf8095f8f335e",

"title":"Squid Access Logs",

"description":"Squid3",

"disabled":false,

"matching_type":"AND",

"stream_rules":[

{

"type":"EXACT",

"field":"type",

"value":"squid3",

"inverted":false,

"description":""

}

],

"outputs":[

],

"default_stream":false

}

],

"outputs":[

],

"dashboards":[

],

"grok_patterns":[

{

"name":"SQUIDURIPORT",

"pattern":"([\\d]+)"

},

{

"name":"SQUIDURIPATH",

"pattern":"([a-zA-Z0-9-._~!$&'()*+,;=:\\/%?]+)"

},

{

"name":"SQUIDURIPROTO",

"pattern":"([a-z]+)"

},

{

"name":"SQUIDURIFRAG",

"pattern":"([a-z0-9-._~!$&'()*+,;=:]+)"

},

{

"name":"SQUIDURIHOST",

"pattern":"([a-z0-9-._~!$&'()*+,;=]+)"

},

{

"name":"SQUIDCONTENTTYPE",

"pattern":"([-\\w]{1,127}\\/[-\\w]{1,127})|(-)"

},

{

"name":"SQUIDURIUSER",

"pattern":"([a-z0-9-._~!$&'()*+,;=:]+)"

},

{

"name":"SQUIDURI",

"pattern":"(?:%{SQUIDURIPROTO:URI_Protocol;string}:\\/\\/|)(?:%{SQUIDURIUSER:URI_User;string}@|)%{SQUIDURIHOST:URI_Host;string}(?:\\:%{SQUIDURIPORT:URI_Port;int}|)(?:\\/%{SQUIDURIPATH:URI_Path;string}|)(?:#%{SQUIDURIFRAG:URI_Frag;string}|)"

}

],

"lookup_tables":[

],

"lookup_caches":[

],

"lookup_data_adapters":[

]

}

Anexo 3

Extractors exportados desde Graylog que pertenecen al inputs Squid-Logs

{
"extractors": [
{
"title": "Squid Extractor",
"extractor_type": "grok",
"converters": [],
"order": 0,
"cursor_strategy": "copy",
"source_field": "message",
"target_field": "",
"extractor_config": {
"grok_pattern": "^%{BASE10NUM:epoch_time}%{SPACE:UNWANTED}%{INT:Duration;int} %{IPV4:Client_Address} %{DATA:Result_Code;string}\\/%{BASE10NUM:Status_Code;int} %{INT:Bytes;int} %{WORD:Method;string} %{SQUIDURI:URI;string} %{DATA:User;string} %{DATA:Peer_Status;string}\\/%{DATA:Peer_Host;string} %{DATA:Content-Type;string}$",

"named_captures_only": false
},
"condition_type": "none",
"condition_value": ""
},
{
"title": "Unix Time Extractor",
"extractor_type": "substring",
"converters": [],
"order": 0,
"cursor_strategy": "copy",
"source_field": "epoch_time",
"target_field": "epoch_time",
"extractor_config": {
"end_index": 10,
"begin_index": 0
},
"condition_type": "none",
"condition_value": ""
},
{
"title": "extractor_su_sucessful_tisw_login",
"extractor_type": "grok",
"converters": [],
"order": 0,
"cursor_strategy": "copy",
"source_field": "message",
"target_field": "",
"extractor_config": {
"grok_pattern": "%{su_sucessful_tisw_login}"
},
"condition_type": "regex",
"condition_value": ".*session opened for user root by*"
},
{
"title": "extractor_ssh_sucessful_login",
"extractor_type": "grok",
"converters": [],
"order": 0,
"cursor_strategy": "copy",
"source_field": "message",

"target_field": "",
"extractor_config": {
"grok_pattern": "%{ssh_sucessful_login}"
},
"condition_type": "none",
"condition_value": ""
},
{
"title": "extractor_su_sucessful_root_login",
"extractor_type": "grok",
"converters": [],
"order": 0,
"cursor_strategy": "copy",
"source_field": "message",
"target_field": "",
"extractor_config": {
"grok_pattern": "%{su_sucessful_root_login}"
},
"condition_type": "none",
"condition_value": ""
},
{
"title": "extractor_admin_access_event-1",
"extractor_type": "regex",
"converters": [],
"order": 0,
"cursor_strategy": "copy",
"source_field": "message",
"target_field": "admin_access_event",
"extractor_config": {
"regex_value": "(?i)(Successful su for root)"
},
"condition_type": "none",
"condition_value": ""
},
{
"title": "extractor_ssh_brute_force_attack",
"extractor_type": "grok",
"converters": [],
"order": 0,
"cursor_strategy": "copy",

"source_field": "message",
"target_field": "",
"extractor_config": {
"grok_pattern": "%{ssh_brute_force_attack}"
},
"condition_type": "none",
"condition_value": ""
},
{
"title": "extractor_ssh_failed_login",
"extractor_type": "grok",
"converters": [],
"order": 0,
"cursor_strategy": "copy",
"source_field": "message",
"target_field": "",
"extractor_config": {
"grok_pattern": "%{ssh_failed_login}"
},
"condition_type": "none",
"condition_value": ""
},
{
"title": "extractor_sudo_auth_failure",
"extractor_type": "grok",
"converters": [],
"order": 0,
"cursor_strategy": "copy",
"source_field": "message",
"target_field": "",
"extractor_config": {
"grok_pattern": "%{sudo_auth_failure}"
},
"condition_type": "none",
"condition_value": ""
},
{
"title": "extractor_su_sucessful_tisw_login",
"extractor_type": "grok",
"converters": [],
"order": 0,

"cursor_strategy": "copy",
"source_field": "message",
"target_field": "",
"extractor_config": {
"grok_pattern": "%{su_sucessful_tisw_login}"
},
"condition_type": "none",
"condition_value": ""
},
{
"title": "extractor_su_sucessful_tisw_login",
"extractor_type": "grok",
"converters": [],
"order": 0,
"cursor_strategy": "copy",
"source_field": "message",
"target_field": "",
"extractor_config": {
"grok_pattern": "%{su_sucessful_tisw_login}"
},
"condition_type": "none",
"condition_value": ""
},
{
"title": "extractor_sudo_sucessful_as_root",
"extractor_type": "grok",
"converters": [],
"order": 0,
"cursor_strategy": "copy",
"source_field": "message",
"target_field": "",
"extractor_config": {
"grok_pattern": "%{sudo_sucessful_as_root}"
},
"condition_type": "none",
"condition_value": ""
},
{
"title": "copy_message_winlogbeat",
"extractor_type": "regex",
"converters": [],

```
 "order": 0,
 "cursor_strategy": "copy",
 "source_field": "message",
 "target_field": "mymessage",
 "extractor_config": {
 "regex_value": "(.+?)Sujeto*"
 },
 "condition_type": "none",
 "condition_value": ""
 }
 ],
 "version": "2.4.3"
}
```


Anexo 4

Ejemplo de mensaje de prueba bajando el umbral de uso de CPU a 40%