Centralización y análisis de eventos de seguridad con Graylog

Autor: Alberto Ruiz Aylagas

Tutor: Pau del Canto Rodrigo

Profesor: Helena Rifà Pous

Master Universitario en seguridad de las tecnologías de la información y las comunicaciones

Análisis de datos

Diciembre 2019

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Centralización y análisis de eventos de seguridad con Graylog*
Nombre del autor:	*Alberto Ruiz Aylagas*
Nombre del colaborador/a docente :	*Pau del Canto Rodrigo*
Nombre del PRA:	*Helena Rifà Pous*
Fecha de entrega (mm/aaaa):	*12/2019*
Titulación o programa:	*Master Universitario en seguridad de las tecnologías de la información y las comunicaciones*
Área del Trabajo Final:	*Análisis de datos*
Idioma del trabajo:	*Español*
Palabras clave	*Graylog, análisis de logs, gestión de eventos de seguridad*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo*

Cada día las empresas tienen más sistemas TIC que generan eventos de seguridad. Analizar estos eventos con editores de texto puede ser muy difícil. Además se tienen que analizar los eventos de cada sistema por separado.

En este trabajo vamos a mostrar la utilidad de utilizar graylog, una herramienta para centralizar el análisis de eventos de seguridad. De esta forma, usando una única herramienta podremos analizar los eventos de varios sistemas de forma sencilla y además mostrar los resultados claramente, pudiendo mostrarlos de manera gráfica.

Para ello utilizaremos dicha solución en un entorno cloud (aws) y realizaremos cuatro casos de uso. Analizaremos eventos SSH, DNS, WAF y firewall.

Finalmente, compararemos esta solución con otras disponibles en el mercado. Concretamente haremos un análisis de Splunk y Eslastic.

Tras finalizar el trabajo habremos comprobado como graylog nos ha permitido realizar los casos de uso de forma sencilla, obteniendo resultados de forma rápida y clara. Además podemos configurar gráficos para tener información disponible en todo momento.

Abstract (in English, 250 words or less):

Every day enterprises have more ICT systems that generate security events. Analizing these events with word processor can be very difficult. In addition, the events of each system must be analysed separately.

In this paper we're going to show the utility of using graylog, a tool to centralize the analysis of security events. In this way, using a single tool we can analyse the events of several systems in a simple way and we can also show the results clearly, being able to show them graphically.

To do this we will use this solution in a cloud environment (aws) and we will perform four use cases. We will analyse SSH, DNS, WAF and firewall events.

Finally, we will compare this solution with others avalaible in the market. Specifically, we will do an analysis of splunk and Elastic.

After finishing this work we will have verified how graylog has allowed us to carry out the use cases easily, obtaining result quickly and clearly. Also, we can configure graphics to have information available in every moment.

Índice

Figuras y tablas

Índice de figuras

Índice de tablas

1. Introducción

1.1. Contexto del trabajo y problema a resolver

Hoy en día, las empresas tienen multitud de sistemas TIC (aplicaciones, elementos de red, servidores, etc) que generan multitud de eventos durante su funcionamiento para mostrar todo lo que está ocurriendo en ellos. Tener un conocimiento preciso de qué es lo que están indicando dichos eventos es algo imprescindible para el correcto funcionamiento de los sistemas, ya que a través de ellos podemos obtener información muy diversa. Por ejemplo, a través de estos eventos podemos tener conocimiento de intentos de autenticación o errores del sistema.

Pero analizar estos eventos para ver que está ocurriendo plantea dos problemas principales:

1- Los eventos se almacenan en ficheros de texto, los cuales pueden contener miles de eventos. Por ello buscar unos eventos concretos puede ser una tarea tediosa, al presentarse toda la información como texto y tener cómo únicas opciones de búsqueda las que nos ofrezca el editor de texto o el sistema operativo, como por ejemplo el comando “grep” de Linux. Pero aún con el uso de estas opciones puede ser complicado realizar búsquedas eficientes.
2- Cada sistema almacena sus eventos de forma local, dentro de sí mismo. Esto plantea la problemática de que si un técnico administra varios sistemas tendrá que acceder a cada uno de ellos para consultar sus eventos.

En este trabajo intentaremos solucionar estos problemas mediante una herramienta de Análisis de eventos, concretamente Graylog. Con Graylog esperamos poder lograr:

1- Integrar los eventos de varias fuentes en Graylog, de forma que no será necesario acceder a cada sistema para consultar sus eventos al tenerlos centralizados.
2- Representar los eventos de manera gráfica mediante cuadros de mando (dashboards).
3- Facilitar la consulta de los eventos al permitir realizar búsquedas mediante filtros, tanto horarios como por el valor que tomen los distintos campos que formen parte de los eventos.

1.2. Objetivos

Los objetivos de este trabajo de fin de master son los siguientes:

1- Configurar Graylog para que procese correctamente los eventos a analizar..
2- Utilizar Graylog para analizar eventos de distintas fuentes y mostrar cómo se pude utilizar dicho análisis para observar datos de interés.
3- Mostrar la información contenida en los eventos a través de cuadros de mando (Dashboards) en Graylog
4- En caso de que los eventos de alguna fuente no sean procesador correctamente por Graylog, realizar un parser que permita procesarlos correctamente.
5- Configurar alertas.

6- Ver qué puntos fuertes y débiles tiene Graylog frente a las soluciones Elastic y Splunk.

Además, como objetivo secundario vamos a tener el comprender cómo funciona un entorno de servidores cloud y aprender a hacer uso del mismo, ya que este trabajo se va a desarrollar en un entorno cloud, concretamente en aws (Amazon Web Services).

1.3. Metodología y proceso de trabajo

Para realizar este TFM se va a utilizar un servicio cloud, concretamente el proporcionado por Amazon (aws) en el que utilizará una de las instancias de Graylog que hay disponibles listas para usar.

Para la realización de este proyecto se va a seguir la siguiente metodología:

En primer lugar, como paso previo a la realización de este trabajo se ha buscado información sobre el tema a tratar en la web, el análisis de eventos de seguridad (logs). Una vez se ha investigado sobre el análisis de eventos se ha elegido el entorno en el que se realizará el proyecto; es decir, dónde se instalará Graylog y se elaborará el plan de trabajo. Posteriormente se buscara documentación tanto del entorno elegido como de Graylog (instalación y configuración) y se procederá a instalarlo y realizar configuraciones de prueba que permitan entender cómo se configura. A continuación, se realizarán los casos prácticos de análisis de eventos. Finalmente se buscara información sobre otras alternativas a Graylog, concretamente Elastic y Spluk, y se realizará una comparativa entre ellas.

Como ya se ha mencionado, se ha valorado en que entorno se realizará el trabajo. Tras valorar las distintas alternativas para usar Graylog se ha decidido a utilizar un servicio cloud, concretamente el proporcionado por Amazon (aws) en el que utilizará una de las instancias de Graylog que hay disponibles listas para usar.

Dado que ya se ha buscado información sobre el tema, ya se ha pensado en algunos casos prácticos de análisis de eventos que se pueden realizar. No obstante, los casos prácticos que se realizarán finalmente se decidirán más adelante puesto que puede ocurrir tanto que se encuentren problemas para realizar alguno de ellos, como no poder obtener los eventos necesarios como que se nos ocurran otros de mayor interés conforme vayamos utilizando Graylog y veamos todas sus posibilidades. Se ha pensado en los siguientes casos prácticos:

1- Analizar los eventos de acceso por SSH para intentar determinar si se ha sufrido un ataque de fuerza bruta a través de los intentos de login fallidos.
2- Análisis de eventos de un servidor DNS para observar cuales son las consultas más frecuentes.
3- Análisis de los eventos de un WAF para intentar detectar actividades anómalas.

Además de esto, durante todo el desarrollo se documentarán las acciones realizadas en la memoria que se entregará al finalizar este TFM, realizándose esta conforme se va realizando el proyecto para que todo quede debidamente documentado.

1.4. Listado de tareas a realizar

Para realizar este TFM se deberán realizar las siguientes tareas:

1- Búsqueda de información sobre el tema a tratar en la web, el análisis de eventos de seguridad.
2- Elección del entorno en que se utilizará Graylog valorando las distintas posibilidades.
3- Elaboración del plan de trabajo
4- Estudio de documentación del entorno en el que se realizará el TFM (aws) y pruebas del mismo
5- Buscar información sobre Graylog (configuración, funcionamiento, etc)
6- Instalar Graylog y realizar configuraciones de prueba.
7- Determinar los casos prácticos de análisis de eventos a realizar
8- Obtener los eventos necesarios para realizar los casos prácticos.
9- Realización de casos prácticos de Graylog,
10- Búsqueda de información sobre otras alternativas a Graylog, concretamente Elastic y Spluk, y comparativa entre ellas.

1.5. Planificación

Para realizar este trabajo se seguirá la siguiente planificación temporal:

Tarea	Fecha de inicio	Fecha de finalización
Búsqueda de información sobre análisis de eventos	18/09/19	22/09/19
Valoración de posibles entornos en los que instalar Graylog y elección del que se utilizará	23/09/19	25/09/19
Elaboración del plan de trabajo	26/09/19	1/10/19
Hito 1 – Entrega del Plan de trabajo		**1/10/19**
Estudio de documentación de aws y pruebas del entorno	2/10/19	8/10/19
Estudio de documentación de Graylog	9/10/19	11/10/19
Instalación de Graylog y realización de configuraciones de prueba	12/10/19	15/10/19
Determinar casos prácticos definitivos a realizar	16/10/19	18/10/19
Obtención de los eventos necesarios para realizar los casos prácticos	19/10/19	22/10/19
Realización de caso práctico 1	23/10/19	29/10/19
Hito 2 – Entrega 2		**29/10/19**
Realización del resto de casos prácticos	30/10/19	26/11/19
Hito 3 – Entrega 3		**26/11/19**
Búsqueda de documentación sobre Elastic y Spluk	27/11/19	3/12/19
Realización de la comparativa de Graylog con Elastic y Spluk	4/12/19	10/12/19
Revisión y finalización de la memoria del TFM	11/12/19	31/12/19
Hito 4 – Entrega memoria TFM		**31/12/19**
Realización de la presentación en vídeo	1/01/20	7/01/20
Hito 5- Entrega del vídeo de presentación		**7/01/20**
Defensa del TFM	13/01/20	17/01/20

Tabla 1: Planificación temporal

1.6. Revisión del estado del arte

Para dar solución al problema descrito en el apartado 1.1, han surgido múltiples herramientas, tanto de pago como gratuitas o con un modelo mixto, que ofrece una versión gratuita limitada en funciones y soporte y otra de pago mucho más completa. Este último sería el caso de Graylog que será la herramienta que utilizaremos en este TFM, en su versión Open Source. Si optáramos por la versión de pago, además de disponer de todas las funcionalidades de la versión Open Source, dispondremos de funcionalidades añadidas entre las que destacan el motor de correlación, la programación de informes y el soporte técnico.

Otras dos herramientas destacadas son Elastic y Splunk. Al final del TFM se estudiarán más en detalle sus características y se compararán con Graylog, pero podemos hacer un breve comentario sobre ellas.

Elastic es un conjunto de herramienta de búsqueda y analíticas con la que podemos realizar el análisis de eventos. Su herramienta principal es Elastic Stack, la cual en realidad son cuatro herramientas que se utilizan de forma conjunta. Concretamente está formada por una herramienta de recolección de logs (Beats), una de ingesta de logs (Logstash), una base de datos no relacional (Elasticsearch) y otra para visualizar los datos (kibana). Al igual que ocurre con Graylog, Elastic Stack permite recibir eventos desde múltiples fuentes, es escalable y se pueden hacer despliegues distribuidos.

Actualmente prácticamente todas las herramientas de Elastic se ofrecen como software open source que podemos utilizar de forma gratuita. No obstante además dispone de tres suscripciones de pago con las que dispondríamos de más funcionalidades en estas herramientas así como de soporte técnico.

Además, también se distribuyen sus herramientas como SaaS de pago. De esta forma, podemos utilizar las herramientas de forma online sin necesidad de tener que instalar ni mantener las mismas.

Splunk es otra herramienta para el análisis de datos. En este caso cuenta con una licencia gratuita de uso personal, pero está limitada a tan sólo procesar 500 MB de datos al día. Además, tiene ciertas limitaciones respecto a la versión Entrerpise, como por ejemplo carecer de un sistema de autenticación para acceder a él o la imposibilidad de generar alertas. El resto de versiones son de pago y dispone tanto de soluciones software tradicionales como soluciones Cloud.

1.7. Recursos necesarios y presupuesto

Para realizar el TFM serán necesarios los siguientes elementos:

1- Un ordenador
2- Conexión a internet
3- Una cuenta en el servicio cloud de amazon en el que desarrollar el trabajo.

El presupuesto para realizarlo, teniendo en cuenta que ya se posee un ordenador y conexión a internet será únicamente el coste del servidor que se utilizará en aws.

Dado que al registrarnos en aws seremos nuevos clientes, dispondremos de una capa gratuita ilimitada disponible para todos los usuarios, así como de una serie de recursos limitados disponibles de manera gratuita por un periodo de 12 meses desde la fecha de registro. De estos recursos, para desarrollar este proyecto nos serán de utilidad los siguientes:

- 750 horas al mes de Amazon EC2 (Elastic Compute Cloud) de instancias t2.micro con Linux, EHEL, SLES o Windows.
- 30 GB de Amazon EBS (Elastic Block Storage)

Posteriormente se entrara más en detalle en estos conceptos en el apartado “Entorno de desarrollo de los casos prácticos”, pero de momento diremos que las instancias Amazon EC2 es una configuración hardware sobre la que se ejecuta el sistema operativo que elijamos. En el caso de la capa gratuita únicamente se nos ofrece la instancia t2.micro, la cual tiene 1 GB de memoria, insuficiente para que funciones Graylog, por lo que será necesario utilizar una instancia de pago. Para lo que si que podemos aprovechar esta instancia gratuita es para la generación de eventos que posteriormente enviemos a Graylog. También hay que señalar que podemos tener varias instancias t2.micro funcionando a la vez, pero el tiempo de uso será la suma de todas las instancias que estemos utilizando.

En cuando a Amazon EBS es el almacenamiento persistente que se nos proporciona, de forma que los datos que guardemos en volúmenes de este tipo no se perderán aunque paremos la instancia, únicamente desaparecerán los datos al terminar la instancia que utilice el volumen si así lo hemos indicado. En caso contrario no los perderemos hasta que eliminemos el volumen de forma manual. Los 30 GB gratuitos que se nos ofrecen serán entre todas las instancias que ejecutemos.

Por tanto, el coste de utilizar aws será el primer lugar el coste de la instancia donde ejecutemos Graylog. Viendo las instancias disponibles, utilizaremos una instancia de tipo t3a.large al contar con 8GB de memoria, suficientes para ejecutar Graylog correctamente. El coste de esta instancia (para la región EU-París, que es la que utilizaremos) es de 0,085$/h más IVA. Este precio se cobra únicamente por uso, de forma que si paramos la instancia no se nos cobrará hasta que volvamos a arrancarla. Previendo que para desarrollar este trabajo serán necesarias 100h de uso de dicha instancia esto supondrá un coste de 8,5$, a los que si sumamos el 21% de IVA (1,785$) nos da un coste de 10,285$

En cuanto al almacenamiento, los 30 GB serán suficientes en un principio, pero conforme avance el trabajo puede que no sean suficientes si hacemos uso de varias instancias para generar eventos. Por tanto, haremos la previsión de que serán necesarios 20 GB adicionales durante un periodo de 2 meses. Hay que indicar que al ifual que ocurre con EC2, el uso de Amazon EBS se cobra por uso, contabilizado en segundos, aunque en este caso haremos la previsión de que se utilizarán 2 meses completos.

El precio de este servicio varía según el tipo de volumen que utilicemos. En este caso haremos uso de volúmenes SSD de uso general el cual tiene un coste de 0,116$ por GB y mes de uso, aunque como se indica el tiempo que lo usamos se mide en segundos y nos cobrarán de forma proporcional. Por tanto, el uso de 20GB durante 2 meses será 0,116*20*2= 4,64$. A esto hay que sumar el 21% de IVA, lo cual supone 0,9744$, por tanto en total serán 5,6144$

Por tanto, el coste de desarrollar este proyecto en aws con la previsión indicada será de 15,8994$ en total.

1.8. Estructura del resto del documento

Este documento consta de otros cuatro capítulos además de este primer capítulo de introducción.

En el segundo capítulo hablaremos sobre graylog, indicando cuál es su arquitectura y mostrando sus elementos y menús más relevantes de los que podemos hacer uso.

En el tercer capítulo mostraremos cuatro casos prácticos de uso, en los que analizaremos con graylog eventos SSH, DNS, de un WAF y de un firewall. Además también mostraremos el entorno en el que se han desarrollado dichos casos prácticos de uso.

En el cuarto capítulo hablaremos sobre otras herramientas que se pueden utilizar para el análisis de eventos de seguridad: Splunk y Elastic

Finalmente, en el quinto capítulo expondremos las conclusiones así como posibles líneas futuras con las que continuar el trabajo.

2. Graylog

Graylog es una herramienta de análisis de eventos Open Source, con una versión Enterprise de pago que añade funcionalidades. Esta herramienta nos permitirá centralizar el análisis de eventos al ser posible recibir eventos de múltiples sistemas. Esto tiene la ventaja de que desde un único lugar podemos ver que está ocurriendo en todos nuestros sistemas, en lugar de tener que acudir al fichero de logs de cada uno de ellos. Además, no sólo podremos recibir los eventos que ocurran en dichos sistemas, si no que también podremos tratarlos, decidiendo cómo almacenarlos, realizar búsquedas, realizar un parseo de los eventos para obtener la información que queramos y crear cuadros de mando donde poder ver diversa información de manera gráfica.

En este capítulo abordaremos la arquitectura de Graylog y su funcionamiento.

2.1. Arquitectura de Graylog

La arquitectura básica de Graylog es la siguiente:

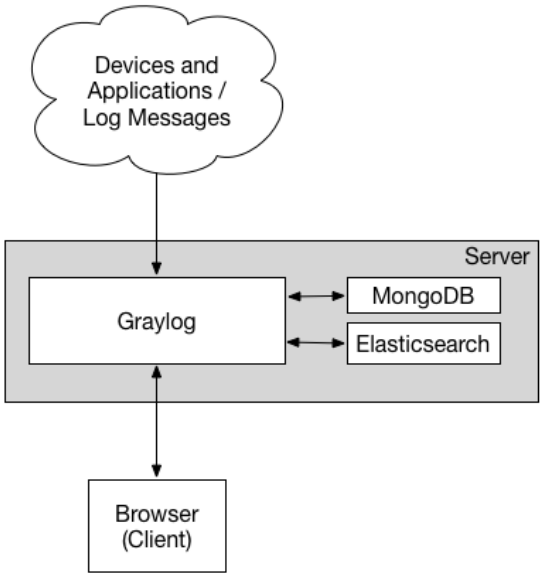


Figura 1: Arquitectura básica de Graylog

Como podemos ver, un servidor con Graylog contiene los siguientes nodos:

- Graylog: Es el nodo central. El será quién reciba los eventos y los procesará. Además es el nodo con el que interactuará el usuario a través de un navegador web.
- MondoDB: Base de datos dónde se almacenarán las configuraciones que hagamos en Graylog, como pueden ser los Streams, reglas, etc.
- ElasticSearch: Almacena los eventos recibidos tras ser procesados por Graylog.

Esta arquitectura es la más básica que nos podemos encontrar, pero Graylog es una aplicación altamente configurable, que nos permite poder desplegar configuraciones más complejas de forma distribuida, dónde estos nodos no estén en el mismo servidor. E incluso, podemos tener alguno de sus nodos replicados y repartiendo su trabajo con un balanceador de carga, ya sea para mejorar el rendimiento o para que sea tolerable a fallos. Un ejemplo de un arquitectura de este tipo sería la siguiente:

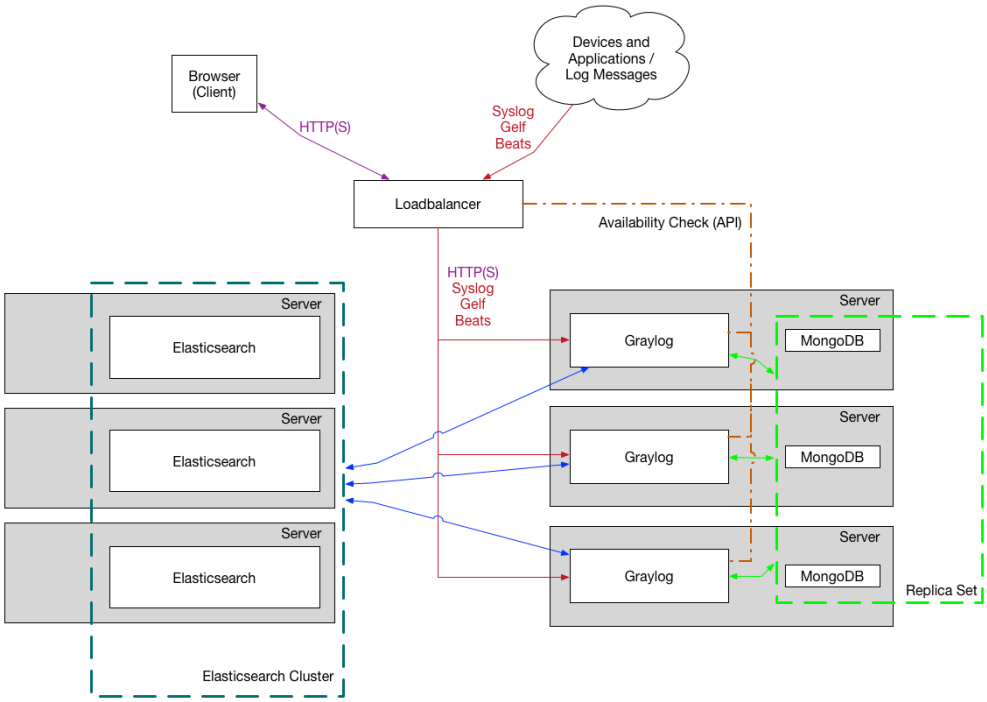


Figura 2: Arquitectura avanzada de Graylog

En este escenario podemos ver como Elasticsearch se ha desplegado de forma independiente del servidor Graylog y MongoDB. Además, se dispone de tres servidores ElasticSearch, lo que mejorará el rendimiento. En el caso de Graylog y MongoDB se mantienen juntos pero también se dispone de tres servidores a modo de réplica.

2.2. Elementos de Graylog

En este apartado pasaremos a mencionar y dar una pequeña explicación de los distintos elementos y menús más relevantes que componen graylog. Además daremos indicaciones sobre como usar algunos de ellos.

La funcionalidad básica que tiene cada uno de estos elementos que mencionaremos en los subapartados siguientes es la siguiente:

- **Input**: Mecanismo utilizado para recibir los mensajes
- **Streams**: Mecanismo para enrutar los mensajes en categorías
- **Search**: Menú a tracés del cual podemos hacer búsquedas entre los eventos recibidos
- **Extractors**: Mecanismo mediante el que podemos parsear los mensajes recibidos para obtener información de ellos.
- **Dashboards**: Menú a través del cual podemos crear y visualizar distintos cuadros de mando
- **Alerts**. Menú en el que podemos configurar alertas, las cuales se activarán en cuando se den las condiciones que indiquemos.
- **Pipeline**: Mecanismo avanzado para tratar los mensajes que reciba graylog.
- **Sidecar**: extractor de eventos que nos permitirá configurar la extracción de eventos de múltiples fuentes de forma centralizada desde Graylog.

2.2.1. Input

Los Input son los mecanismos por los que Graylog puede recibir los mensajes. Entre otros mecanismos, se encuentran:

- Syslog (TCP, UDP, AMQP, Kafka)
- GELF (TCP, UDP, AMQP, Kafka, HTTP)
- Beat/Logstash
- CEF (TCP, UDP, AMQP, Kafka)
- JSON

Graylog nos permite crear tantos Input como queramos y podemos en todo momento elegir cuales estarán funcionando y cuales detenidos. Al crear un Input podemos configurar varios parámetros, como el puerto en el que escuchara el Input o si queremos cifrar los datos mediante TLS haciendo uso de certificados digitales. Los Input que creemos se pueden iniciar y parar en cualquier momento.

Una vez creado un Input deberemos iniciarlo y configurar el envío de los mensajes en el emisor mediante el método elegido. Por ejemplo, si hemos creado un Input de tipo Beat y los mensajes se enviarán desde un servidor Linux deberemos instalar en este filebeat y configurarlo adecuadamente para que envíe los mensajes a Graylog.

Podemos encontrar la configuración de los Input en System/Input

2.2.2. Streams

Los Streams son un mecanismo para enrutar los mensajes recibidos por Graylog en categorías en tiempo real. Este enrutamiento se hace mediante la evaluación de reglas, de forma que cada vez que llega un mensaje se evaluará si cumple las reglas de algún stream, en cuyo caso se enrutará a él.

Hay que tener en cuenta que un único mensaje puede ser enrutado a varios streams. Por ejemplo, podemos tener un stream para todos los eventos relacionados con SSH y otro stream donde se almacenen únicamente los intentos de login fallidos. De esta forma, un log que nos indique un intento de login fallido sería enviado a los dos streams.

La utilidad de tener estos streams es que graylog nos permite ver los mensajes que contiene cada uno de ellos, de forma que cuando los estemos visualizando si realizamos búsquedas o generamos gráficos serán únicamente sobre los mensajes que contenga dicho stream. De esta forma, podremos tener bien categorizados los mensajes y hacer más fácil su tratamiento.

Cuando instalamos Graylog tendremos el stream por defecto llamado “All Messages” al que se enrutaran todos los mensajes recibidos. En caso de crear nosotros nuestros propios Streams podemos elegir si los mensajes que se envíen a dichos streams se deben enviar también o no a este stream.

En la siguiente imagen podemos ver el menú de Streams con el stream por defecto y uno para los mensajes relacionados con SSH.

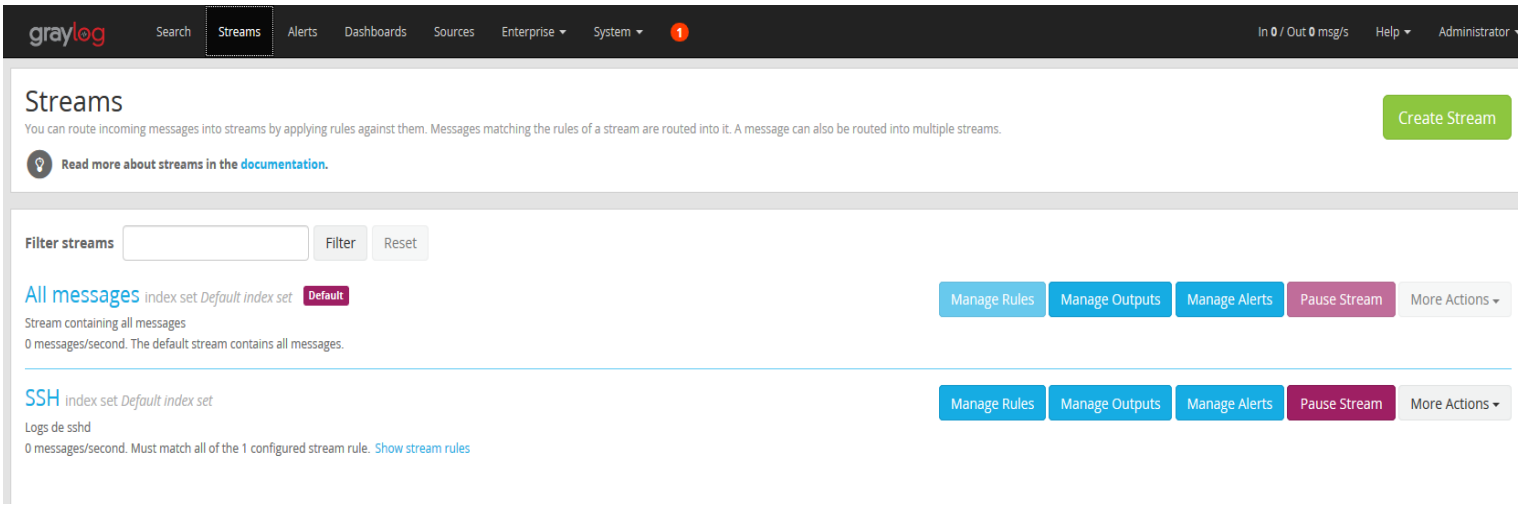


Figura 3: Menú Streams

Para crear un stream simplemente debemos pinchar sobre “Create Stream”, indicar un nombre, descripción, en que Index se escribirán los mensajes de dicho Stream y si queremos que los mensajes también se envíen al stream por defecto.
Una vez creado debemos ir a “Manage Rules” donde estableceremos sus reglas, eligiendo en primer lugar si para que un mensaje se envíe al stream es necesario que cumpla todas las reglas o solo alguna de ellas. Tras esto, debemos crear las reglas, eligiendo en primer lugar el campo que se evaluará y que tipo de evaluación se hará. En la siguiente imagen podemos ver los tipos de evaluación disponibles:

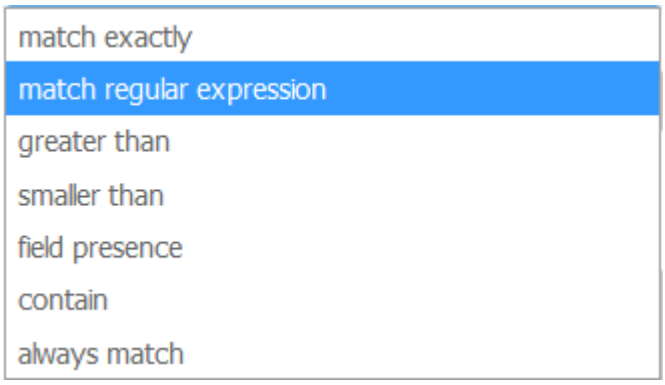


Figura 4: Tipos de evaluación

Finalmente, debemos decir con qué valor se evaluará. En la siguiente imagen podemos ver una regla que comprobará si en el campo “message” contiene la expresión regular “.sshd.” Una vez configurado el stream podemos cargar cualquier mensaje recibido y ver si con las configuración dada entraría o no en dicho stream. Un dato importante a tener en cuenta es que se enrutaran al stream los mensajes recibidos una vez este creado y funcionando, pero no será posible enviar al stream los mensajes recibidos antes de crearlo

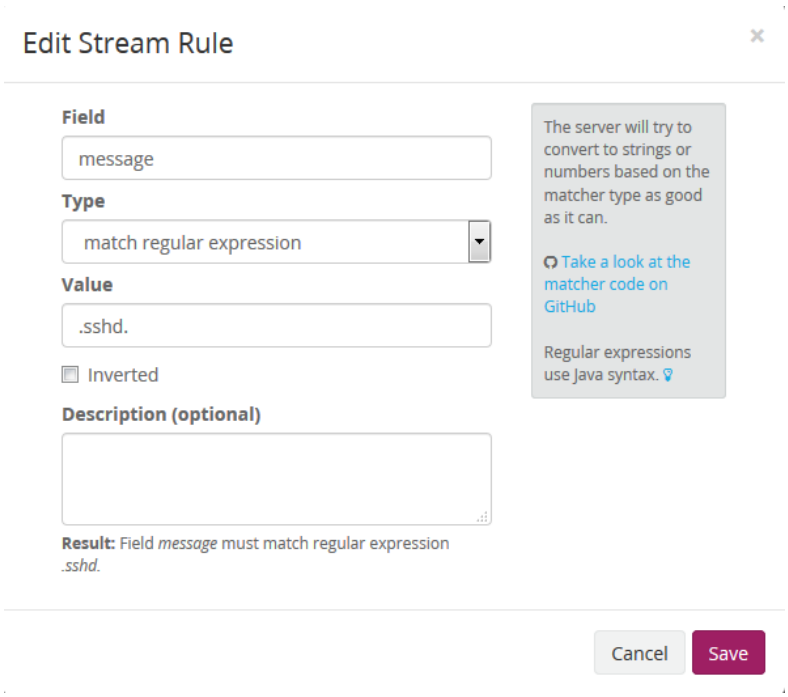


Figura 5: Regla SSH

Al igual que sucede con los Input, podemos iniciar o detener los streams en cualquier momento.

2.2.3. Search

Realizar búsquedas entre los eventos recibidos es una de las funciones principales de graylog. Al entrar en el menú “search” podemos realizar búsquedas sobre los mensajes almacenados en el stream por defecto (“All Messages”). Si queremos realizar búsquedas sobre los mensajes de un stream concreto, debemos entrar en el mismo a través del menú “Streams”.

Las búsquedas se hacen con una sintaxis muy similar a Lucene, por lo que podemos hacer búsquedas que nos devuelvan mensajes que tengan un campo con un valor concreto, mensajes que incluya alguno o todos los términos de una lista, que cumpla con una expresión regular, etc.

Podemos elegir en que intervalo temporal hacer las búsquedas, que puede ser relativo (como por ejemplo los últimos 7 días) o absoluto, en el que elegiremos la fecha y hora del inicio y final del intervalo en el que se hará la búsqueda.

Para facilitar el no tener que realizar constantemente las mismas búsquedas podemos guardar una búsqueda, de esta forma se guardarán tanto los términos de la búsqueda como el intervalo temporal seleccionado bajo el nombre con el que queramos guardarla. Así, podemos repetir la misma con tan solo seleccionarla de una lista, sin necesidad de volver a configurarla.

Una vez realizada una búsqueda, además de mostrarnos los mensajes resultado de la misma, se nos mostrará un histograma en el que podemos ver el resultado de forma gráfica. En este histograma podemos elegir como visualizar los gráficos, si por días, semanas, horas, etc.

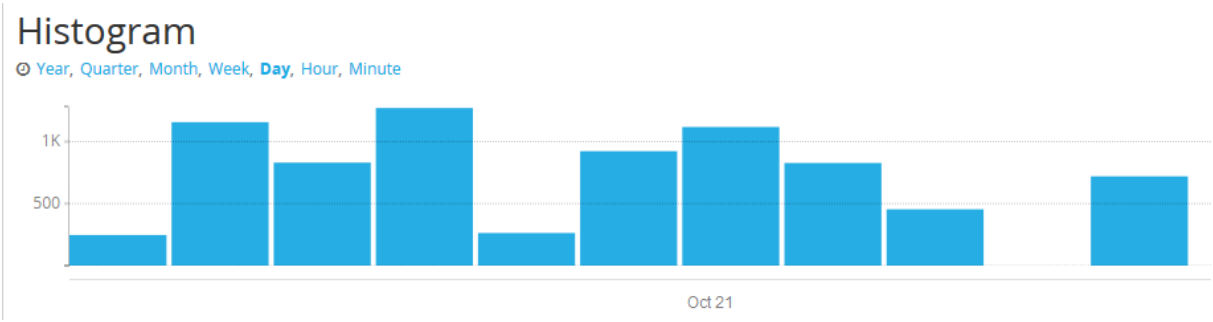


Figura 6: Histograma por días de una búsqueda

Además, podemos añadir información, tanto gráfica como estadística, sobre cualquier campo de los mensajes resultado de la búsqueda y añadir los mismo a algún cuadro de mando en el apartado Dashboards.

2.2.4. Extractors

Los extractores nos permitirán parsear los mensajes recibidos por graylog, es decir, podremos obtener datos de cualquier mensaje recibido por graylog sin importar su formato y además, los datos extraídos se almacenarán en un campo de forma que podamos hacer búsquedas sobre ellos. Esto es algo muy útil, ya que graylog no siempre será capaz de obtener por si solo dentro de campos toda la información que necesitemos, pero configurando un extractor podemos obtenerla.

Los extractores hay que definirlos para cada Input, de forma que si tenemos varios input no tienen por qué usar los mismos extractores.

Los extractores se pueden configurar de varias formas, por ejemplo mediante expresiones regulares, patrones Grok o JSON. Al igual que con los streams, podemos probar el extractor con cualquier mensaje recibido. En este caso también se aplicará únicamente a los mensajes recibidos tras su creación, no es posible aplicar un extractor a los mensajes recibidos con anterioridad.

2.2.5. Dashboards

Este menú nos permitirá crear distintos cuadros de mando donde podamos ver gráficos y estadísticas sobre determinadas búsquedas. De esta forma, podemos observar de forma gráfica aquella información que nos pueda ser relevante, como podría ser el número de veces que se loguea cada usuario en un sistema, el número total de excepciones recibidas en un día o un ranking de los ataques detectados por un IDS.

2.2.6. Alerts

Graylog permite la generación de alertas, las cuales se pueden enviar por correo electrónico o a otro sistema a través de HTTP. En este menú podemos configurar eventos, los cuales se activan cuando se cumplen una serie de condiciones. Una vez se activa el evento se produciría la alerta.

En el caso de Graylog Enterprise además es posible combinar eventos para hacer correlaciones.

2.2.7. Pipeline

Los pipeline son un mecanismo complejo para tratar los mensajes que llegan a graylog. Mediante un pipeline podemos decidir hacia dónde enrutar los mensajes así como añadirles información adicional.

Al contrario que sucede con los streams, donde simplemente se establecen unas reglas y se define si se debe cumplir alguna de ellas o todas, los pipeline constan de una serie de etapas. De esta forma, en cada etapa se puede elegir si para pasar a la siguiente (o ser entrutado a un stream) se deben cumplir

todas o alguna de las reglas. Además, las reglas pueden ser mucho más complejas que las de los streams permitiendo incluso el uso de funciones escritas en Java. Por ello, este es un mecanismo de enrutamiento y tratamiento de mensajes mucho más potente que los streams.

2.2.8. Sidecar

Como se mencionó en el apartado 1.2.1, cuando creamos un Input es necesario configurar el método de envío de eventos en cada una de las fuentes. Esto puede ser algo laborioso en caso de disponer de muchas fuentes, ya que deberemos configurar el envío de eventos en todas ellas. Por ello Graylog nos ofrece un colector de logs llamado Sidecar.
Con Sidecar podemos centralizar la configuración del envío de eventos a graylog. Basta con instalar Sidecar en los equipos que enviarán eventos a graylog. Una vez hecho, desde la consola web de graylog podremos configurar el método de envío de eventos que queramos utilizar.

Por defecto, Sidecar ya incluye la configuración de filebeat y nxlog para Linux y winlogbeat y nxlog para Windows, de forma que si usamos alguno de estos métodos para enviar los eventos a graylog tan solo deberemos editar su fichero de configuración desde el apartado Sidecars. Pero en caso de querer utilizar otro método para enviar los eventos es posible añadir la configuración a Sidecar.

Una vez hemos instalado Sidecar en los equipos, estos nos aparecerán en la consola web y podemos aplicar a cada uno la configuración que deseemos. Esto tiene la ventaja, además de no tener que acudir a cada equipo de forma individual como ya se ha dicho, que si un gran número de equipos van a usar la misma configuración tan solo tendremos que hacerla una vez y se la podemos aplicar a todos, sin necesidad de repetirla una y otra vez.

3. Casos prácticos de uso

En este capítulo abordaremos distintos casos prácticos de uso de graylog que podrían darse en un entorno empresarial para mostrar su utilidad. En primer lugar daremos una explicación sobre el entorno en que se ha llevado a cabo este trabajo. A continuación, pasaremos a realizar los casos prácticos. Para cada uno de los casos daremos en primer lugar una introducción al mismo, para seguir con un subapartado dónde explicaremos como se ha llevado a cabo, es decir, cómo se han enviado los eventos a graylog, y como se han tratado los eventos (que extractores y streams se han configurado y cómo, etc). Finalmente, tendremos un subapartado de resultado, dónde, partiendo de que ya tenemos los eventos en graylog y hemos podido extraer de ellos la información que creemos relevante mostramos como podemos visualizarla para obtener datos de interés.

3.1. Entorno de desarrollo de los casos prácticos

Para realizar estos casos prácticos se ha utilizado el servicio cloud de Amazon (aws) dónde se ha instalado graylog. Usar este servicio cloud nos proporciona las siguientes ventajas:

- No tendremos que preocuparnos por las limitaciones Hardware. Durante este proyecto, si se realizara de forma local sería necesario el uso de varias máquinas virtuales de forma simultánea. Por ello, sería necesario tener memoria suficiente para que funcionen de forma fluida, por lo que al realizarlo en un servicio cloud no nos tendremos que preocupar de ello, simplemente deberemos utilizar servidores con memoria suficiente.
- Nos facilita la obtención de logs. Por ejemplo, si queremos utilizar los logs que se generan al acceder a una aplicación o servicio (como SSH) podemos dejarlo expuesto a internet el tiempo que queramos para registrar los intentos de acceso que se produzcan. En caso de hacerlo de forma local tendríamos que dejar expuesto nuestro ordenador personal, con los riesgos y gasto de electricidad que puede entrañar.
- Finalmente, desarrollar este proyecto en un entorno cloud nos servirá para aprender como funcionan dichos entornos.

Este servicio cloud funciona mediante dos elementos clave que definiremos a continuación: instancias y AMI

- Según la propia documentación de aws “una instancia es un servidor virtual en la nube”. Cada tipo de instancia dispone de una configuración específica de CPU y memoria.
- Por otro lado, una AMI (Imagen de máquina de Amazon) es, según se indica en la documentación de aws, “una plantilla que contiene una configuración de software (por ejemplo, un sistema operativo, un servidor de aplicaciones y aplicaciones)”.

Es decir, que una AMI siempre se ejecutará sobre una instancia, y al arrancar esta instancia lo hará con ejecutando el sistema operativo que contenga la AMI. Obviamente, no hay una relación entre las AMI y las instancias. Una misma AMI puede funcionar en cualquier tipo de instancia, siempre y cuando esta cumpla los requisitos hardware que tenga su software para poder funcionar. E incluso podemos lanzar una misma AMI en varias instancias, sean del mismo tipo o no. Podemos ver esto en el siguiente esquema:

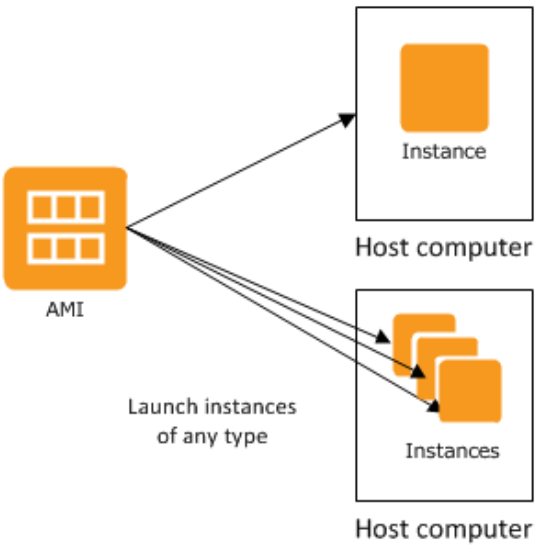


Figura 7: AMI e instancias

Para realizar este trabajo se ha utilizado una instancia de tipo t3a.large, el cual tiene 2 vCPU y 8GB de memoria RAM.

En cuanto a la AMI se ha utilizado una proporcionada con Graylog, la cual es un sistema Ubuntu con Graylog ya instalado y configurado, listo para acceder a su consola web y empezar a utilizarlo. La AMI concreta es la siguiente: **graylog-3.0.2-1** - ami-0494f102259289d85.

Una vez arrancada la instancia que contiene Graylog, si accedemos a la misma mediante SSH se nos mostrará cual es la contraseña para poder acceder a través de la consola web con el usuario admin. Por motivos de seguridad he cambiado tanto este nombre de usuario como su contraseña. Esto no se puede hacer desde la consola web, si no que hay que hacerlo editando el fichero /etc/graylog/server/server.conf donde en root_username podemos cambiar este nombre de usuario y en root_password_sha2 su contraseña (la cual hay que introducir cifrada). Además también se ha cambiado el valor de password_secret, el cual se utiliza para el salt de las contraseñas cifradas.

Durante el desarrollo del trabajo se ha observado que una vez recibido un evento (log), no es posible moverlo de stream si no se ha enrutado correctamente ni aplicarle un extractor y además no es sencillo eliminar los logs recibidos. Por ello se ha decidido trabajar con dos instancias de Graylog, una para realizar pruebas, y ver si los mensajes se enrutan correctamente y si se aplican bien los extractores. Una vez se ha visto cual es la configuración correcta se ha hecho esta misma configuración en la segunda instancia de graylog y se han mandado los eventos. De esta forma se ha podido trabajar sobre ellos, realizando búsquedas y creando cuadros de mando, sin tener una serie de eventos que no se vana estar tratando correctamente.

Para el envío de los eventos, así como para generar los eventos de SSH se ha utiliado la AMI:

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-087855b6c8b59a9e4

Sobre instancias de tipo t2.micro las cuales forman parte de la capa gratuita.

Finalmente, hay que tener en cuenta que las instancias pueden tener su dispositivo raíz de uno de dos tipos: con respaldo en el almacén de instancias o respaldadas por Amazon EBS.

Si el dispositivo raíz utiliza respaldo en el almacén de instancias esto significa que los datos persisterán únicamente mientras la instancia este en funcionamiento, pero si esta se detiene por algún motivo los datos se perderán.

Por el contrario, si la instancia utiliza como dispositivo raíz un volumen de Amazon EBS, los datos persistirán hasta que se finalice la instancia o se elimine el volumen manualmente si así se ha decidido al crearlo. Pero en caso de que simplemente se pare la instancia no se perderán los datos. Por este motivo se ha buscado que las instancias utilizadas utilicen este tipo de dispositivo raíz, para de esta forma pode parar las instancias cuando no se utilicen sin perder los datos.

3.2. Caso práctico 1. Análisis de eventos SSH

En este escenario vamos a analizar los intentos de login SSH fallidos. Esto puede ser interesante ya que es frecuente permitir el acceso a los servidores mediante SSH.

Para realizarlo, se han iniciado tres instancias distintas de tipo t2.micro en aws con Ubuntu, las cuales se han dejado durante varias horas con el puerto 22 expuesto a internet para recoger intentos de login fallidos. De esta forma, en el fichero /var/log/auth.log han quedado registrados estos intentos de login en los que podemos ver la dirección IP desde la que han intentado hacer login y con qué nombre de usuario.

3.2.1. Desarrollo del caso práctico 1

Una vez obtenidos los eventos de los intentos de login, se ha determinado como hacer el envío de los mismos a graylog. Para ello se ha optado por utilizar un Input de tipo Beat y administrar su recolección mediante Sidecar. De esta forma, tan sólo ha sido necesario instalar Sidecar y filebeat en cada una de las instancias para poder administrar filebeat desde la consola web de graylog. El proceso de configuración de sidecar se muestra en el Anexo A.

Para poder tratar adecuadamente los eventos recibidos se ha decidido crear un Stream llemado SSH en el que se enrutan todos los mensajes relativos a SSH, usando como única regla que el campo "message" contenga la expresión regular "**.sshd.**".

Lo siguiente que se ha hecho ha sido crear dos extractores. Cuando un usuario intenta hacer login con un usuario que no es válido se van a crear dos eventos en el fichero de logs; uno en el que se informa de que el usuario no es válido y otro en el que se indica que se ha cerrado la sesión porque el usuario no es válido. Estos eventos tienen el siguiente formato:

Oct 24 21:52:58 ip-172-31-32-41 sshd[1783]: Invalid user pi from 78.212.88.184 port 48460
21:52:58 ip-172-31-32-41 sshd[1783]: Connection closed by invalid user pi 78.212.88.184 port 48460 [preauth]

Por lo que en los mensajes podemos ver que aparece tanto el nombre de usuario introducido como la dirección IP desde la que se ha intentado hacer login. Para extraer el nombre de usuario hay que crear un extractor. Para ello vamos al Input utilizado (de tipo beat) y creamos un extractor llamado Invalid User que almacenará el nombre de usuario en una variable que llamamos invalid_user. Para ello usamos la expresión regular: *Connection closed by invalid user (.+?)*\s.*

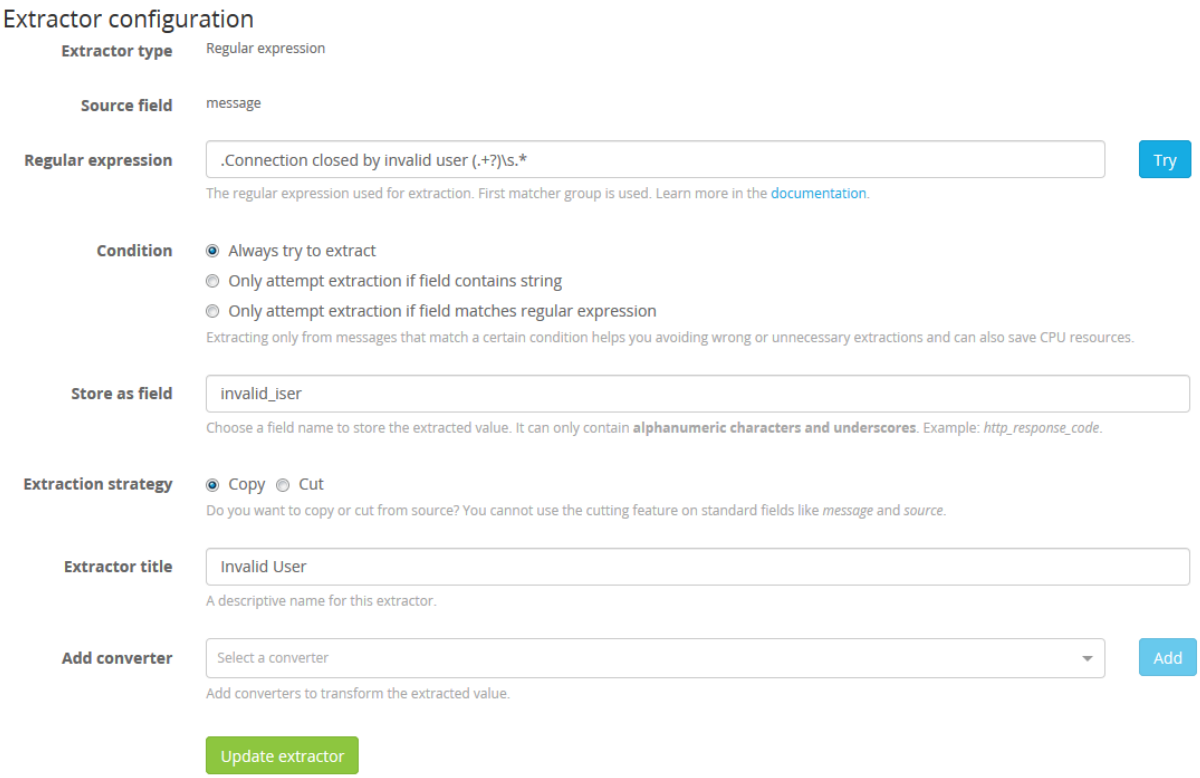


Figura 8: extractor invalid_user

Esta expresión regular se refiere a el segundo evento que se registra, dónde se cierra la sesión, pero de forma análoga podríamos haber extraído el nombre de usuario desde el primer evento con la siguiente expresión regular: *Invalid user (.+?)*\s.*

Es importante usar sólo un extractor (el del segundo evento como hemos hecho, o el del primero como acabamos de indicar), pero no ambos. En caso de que extralleramos el nombre de usuario de ambos eventos lo tendríamos por duplicado y no tendríamos así datos fiables a la hora de crear gráficas y estadísticas.

Para extraer la dirección IP he usado un patrón GROK ya que por defecto Graylog incluye varios patrones, entre ellos uno para obtener una dirección IPv4 el cuál he usado. He llamado a este extractor IPv4.

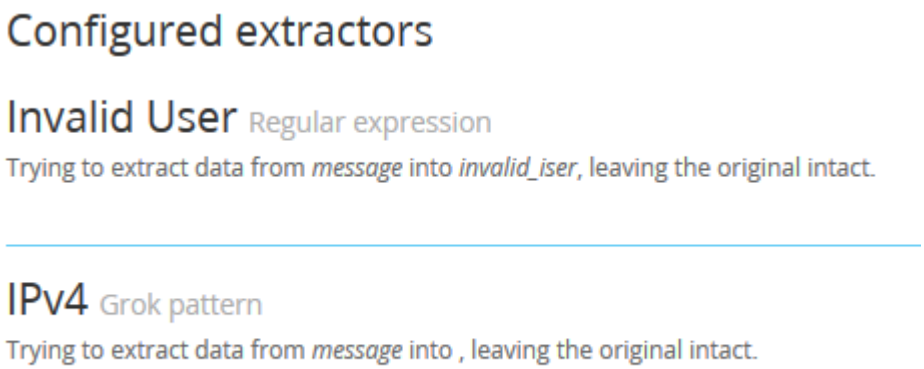


Figura 9: Extractores SSH configurados

Una vez configurados los extractores he entrado en el Stream para ver todos los mensajes recibidos. Si bien todos son sobre SSH no todos son intentos de login fallidos que es lo que interesa en este caso. Por ello he realizado la siguiente búsqueda en los últimos 7 días: *message: "connection closed by invalid user"* y para poder utilizarla posteriormente la he salvado.

Tras realizar esta búsqueda se nos muestra un histograma con el que podemos ver de forma gráfica los intentos de login fallidos recibidos cada día:

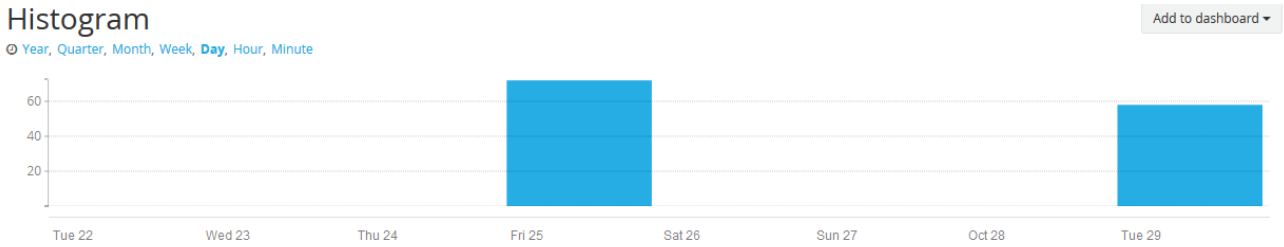


Figura 10: Histograma de intentos de login fallidos

Como vemos los intentos se concentran en un par de días que son los que se han tenido accesibles a internet las instancias de Ubuntu.

3.2.2. Resultado del análisis de eventos SSH

Tras crear los extractores y streams adecuados podemos pasar a consultar los eventos y crear cuadros de mando con el que poder ver la información relevante de forma rápida y gráfica. Para ello creamos un cuadro de mando dentro de “Dashboard” llamado “intentos de login SSH fallidos dónde incluimos la siguiente información relativa a los últimos 7 días:

- Un contador con el número de intentos de login fallidos.
- Un gráfico que muestra la el número de intentos de login fallidos de cada instancia Ubuntu.

- Un gráfico que muestra las IP origen desde las que se han recibido los intentos de login fallidos.
- Un gráfico que indica con qué usuario se ha intentado hacer login.

Los gráficos además cuentan con una tabla que ilustra los resultados en forma de ranking, dando para cada resultado su cantidad y el tanto por ciento que supone.

Añadimos esta información desde la búsqueda realizada anteriormente, dónde hemos buscado cada uno de los campos relevantes (Invalid User, IPv4 y filebeat_source) y he creamos un gráfico “quick values” y posteriormente añadimos los mismos al cuadro de mando.

El cuadro de mando resultante es el siguiente

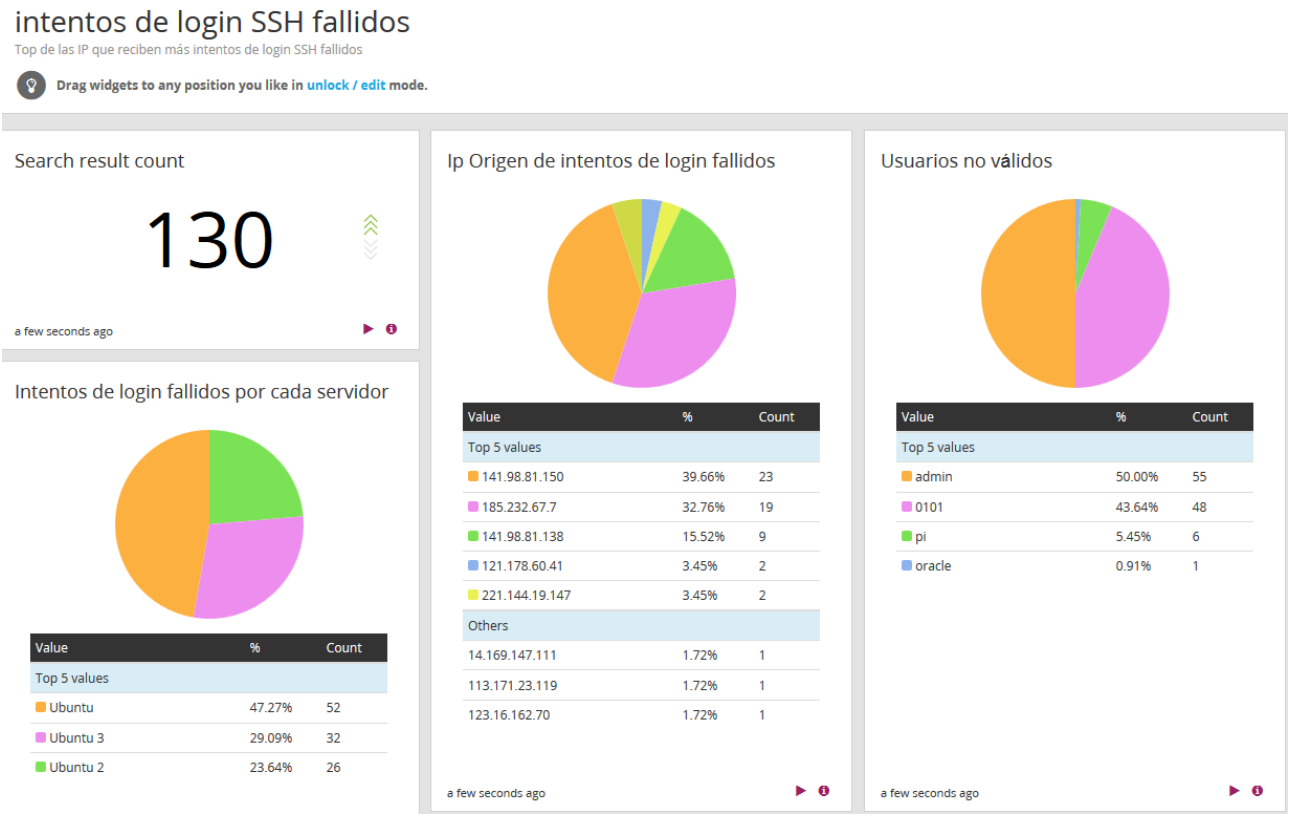


Figura 11: Cuadro de mando “intentos de login SSH fallidos”

3.3. Caso práctico 2. Análisis de logs DNS

En este apartado vamos a analizar los logs de consultas que se generan en un servidor DNS. Esto es interesante ya que en primer lugar nos puede aportar un valor estadístico sobre que dominios se están consultando desde nuestra organización.

Pero además, podría servirnos para tener indicios de que algún equipo está infectado por malware. Esto se debe a que determinado malware, particularmente el asociado a Botnets, necesitan realizar consultas DNS para averiguar en qué IP está el servidor C&C, . Entre otros indicativos, según INCIBE-CERT en su artículo "Detección de Botnets basada en DNS", las siguientes situaciones anómalas pueden ser indicativas de que un equipo esta infectado con el malware de una botnet:

"Búsqueda de nombres de dominio cuya tasas de consulta son anormalmente altas o que están temporalmente concentradas".

También que una misma IP realice un número de consultas DNS anormalmente alto, podría ser indicativo de que dicho equipo está infectado por malware. Detectar esta situación nos debería servir para, al menos, comprobar que dominios está consultando. Según su estructura podrían ser dominios asociados a una botnet, ya que según el artículo de INCIBE-CERT antes mencionado, podemos realizar un *"análisis de las direcciones que se solicitan para su resolución, analizando posible patrones, porcentaje de caracteres numéricos o la inclusión de palabras reconocibles, ya que los atacantes no están preocupados de que sus nombres de dominio sean fáciles de recordar.".*

Además, un número muy elevado de peticiones DNS puede deberse a que estas estén fallando, lo que también según dicho artículo puede ser un indicativo para detectar equipos que forman parte de una Botnet.

Estos indicativos de que un equipo pueda formar parte de una botnet (salvo el último, ya que en los logs disponibles no podemos ver las consultas que han fallado), los vamos a poder observar en Graylog con un tratamiento correcto de los logs DNS. Además, haremos que se generen alertas que recibiremos por correo en dos situaciones: cuando se haga una consulta DNS sobre un dominio malicioso y cuando el número total de consultas DNS exceda un umbral determinado. También mostraremos gráficas sobre el número de consultas que se están realizando y sobre qué dominios se consultan.

3.3.1. Desarrollo del caso práctico 2

Para realizar este caso práctico se usarán logs de query de un servidos DNS facilitados por el tutor de este trabajo, los cuales enviaremos a graylog a través de Sidecar mediante filebeat.

Los eventos analizados tienen el siguiente formato:

04-Nov-2019 10:54:48.908 client @0x7fda8c23f4a0 10.8.102.16#63943 (v10.vortex-win.data.microsoft.com): query: v10.vortex-win.data.microsoft.com IN A + (10.10.0.11)

De los mismos nos van a interesar dos datos, la dirección IP que está realizando la consulta y la consulta que está realizando. Por tanto vamos a crear 2 extractores en el Input “beat” que ya tenemos creado. El de la dirección IP no hace falta crearlo ya que lo tenemos del caso práctico anterior. Aunque en el log aparezcan dos direcciones IP la que nos interesa es la primera, que es la que hace la consulta, por lo que podremos usar ese extractor sin problemas ya que al usar graylog el extractor se quedará con el contenido del mensaje que cumpla el patrón en primer lugar. Es decir, al llegar a la primera dirección IP y ver que cumple el patrón la extraerá y almacenará en la variable llamada IPv4, sin llegar a evaluar la segunda dirección IP.

Para extraer las consultas utilizaremos un extractor que, mediante la expresión regular “query:\s(.+)\s(IN|CH|HS)” extraerá la consulta realizada y la almacenará en la variable query:

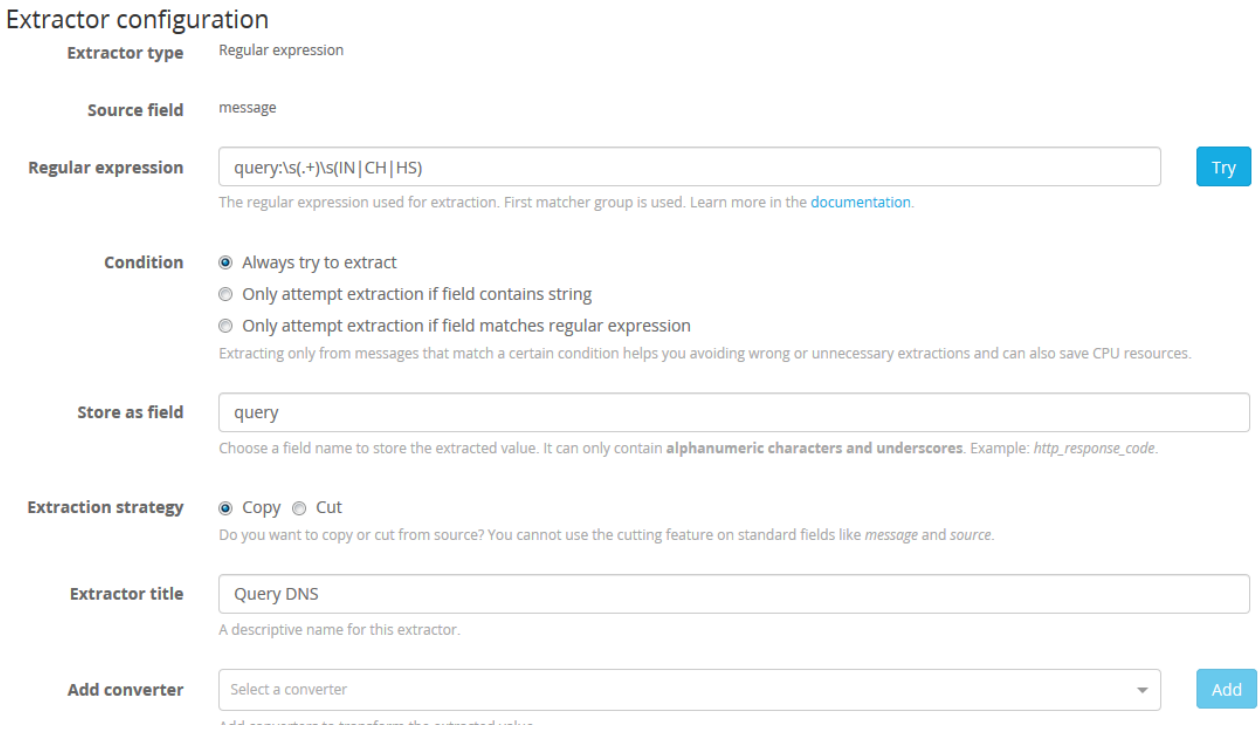


Figura 12: estractor Query DNS

Finalmente, para tener correctamente clasificados los eventos recibidos vamos a crear un Stream llamado DNS dónde se almacenarán todos los eventos de consultas DNS recibidos. Para ello utilizaremos una única regla que tendrán que cumplir los mensajes, la cual consiste en la expresión regular .query. En la siguiente imagen podemos ver la configuración de la regla de este stream:

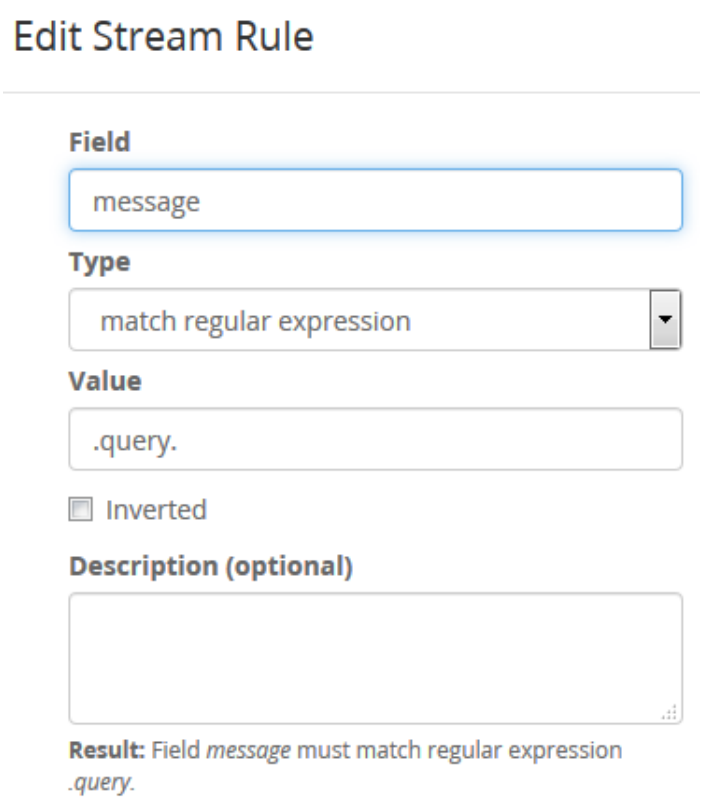


Figura 13: Stream DNS

3.3.2. Resultado del análisis de eventos DNS

Una vez tenemos en graylog los eventos DNS almacenados dentro del stream DNS y hemos extraido de ellos la información relevante vamos a pasar a tratarla. Para ello nos vamos al stream y visualizamos los eventos de la última hora. Una vez los estemos visualizando vamos a crear dos gráficos: Uno con un ranking de las IP que están realizando consultas DNS y otro con un ranking de las consultas realizadas. Además, vamos a crear un contador que nos muestre el número total de peticiones DNS realizadas. Incluimos toda esta información en un cuadro de mando llamado DNS, al cual podemos acceder al través del menú “Dashboards”. El cuadro de mando resultante es el siguiente:

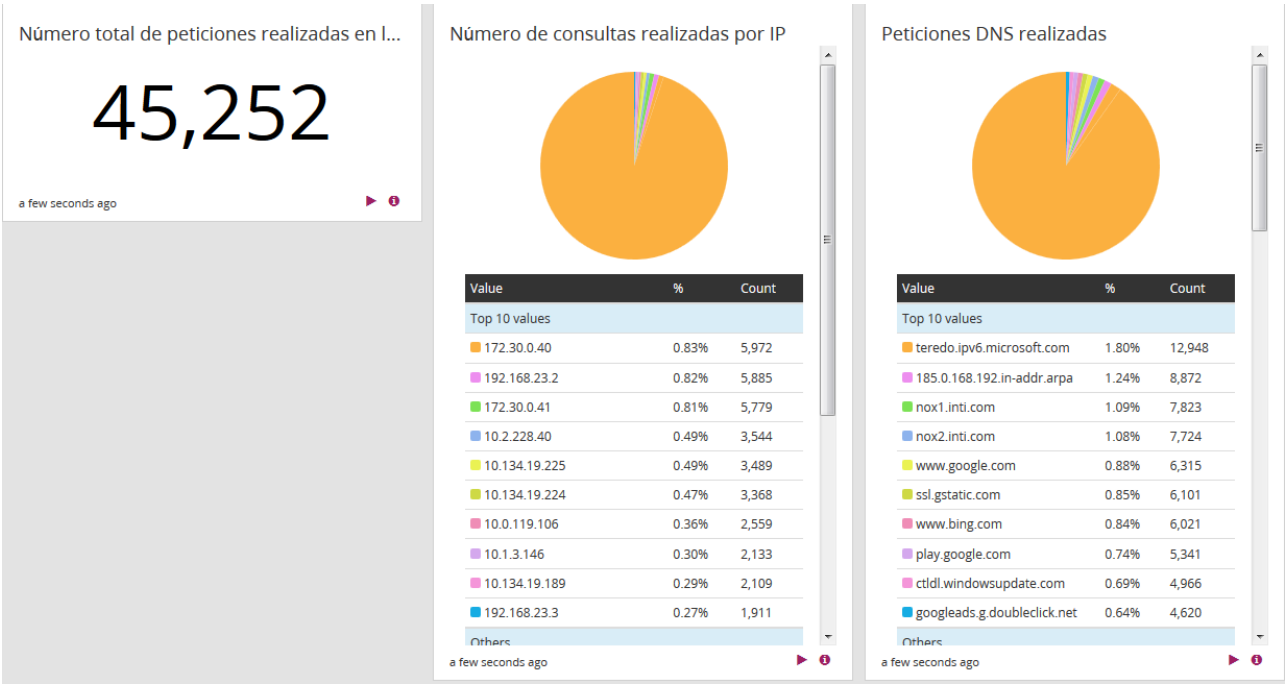


Figura 14: cuadro de mando DNS

Como podemos observar se muestra la información que habíamos dicho. Hay que mencionar que en este caso no estamos trabajando con eventos recibidos en tiempo real, por lo que la información mostrada no varía. Pero si se estuvieran recibiendo los eventos de un servidor DNS en tiempo real si que cambiaría la información que visualizaríamos, mostrándose siempre datos de la última hora.

Como podemos ver, a través de los gráficos podríamos detectar de forma visual que una dirección IP estuviera realizando una cantidad de consultas DNS anormalmente alta, o que se estén realizando un gran número de consultas sobre un dominio. Como además podemos ver los dominios por lo que se está preguntando podríamos ver alguno cuyo nombre sea sospechoso de pertenecer a una botnet.

En cuanto al contador nos permite saber cuál es el número de consultas realizadas en la última hora, aunque podríamos hacer que nos mostrara el dato de las últimas 24h. De esta forma si viéramos que el número que se muestra es anormalmente alto comparado con el que hay habitualmente podría ser un indicativo de que está sucediendo algo “raro” y habría que investigar que está ocurriendo.

3.3.3. Configuración de alertas DNS

Una vez hemos analizado los logs DNS vamos a pasar a configurar alertas que se activarán en dos situaciones: cuando el número de consultas DNS sobrepase un umbral determinado o cuando se haga una consulta DNS sobre un dominio malicioso (asociado por ejemplo a la actividad de un malware).

Para configurar estas alertas debemos ir “Alerts” y una vez aquí en la sección “Conditions” podemos configurar las distintas condiciones que pueden hacer que se active una alerta. Hay que mencionar que en la versión gratuita de graylog, no es posible establecer una lógica entre las distintas condiciones que tengamos configuradas (por ejemplo que se deban cumplir dos condiciones para activar la alerta) si no que se generará una alerta cada vez que su cumpla alguna condición.

Una vez estamos en la sección “Conditions” vamos a pasar a configurar la primera condición, que la consulta se haga sobre un dominio malicioso. Para ello creamos una condición nueva y veremos que se nos solicitan dos datos:

- El stream en el que se puede dar la condición que activará la alerta. Es decir, sólo se examinará la condición para el stream que seleccionemos, que en este caso será el stream DNS. Podemos elegir el stream por defecto (“All messages”) pero no podemos elegir dos streams simultáneamente. Por lo tanto si una condición se puede dar con los mensajes que reciban dos streams distintos no quedará más remedio que o bien seleccionar el stream por defecto o bien crear la condición dos veces, una para cada stream.
- Elegir el tipo de condición que se usará. Se nos ofrecen tres posibilidades, que la condición se cumpla con el contenido de un campo (tenga un valor concreto), por la cantidad de mensajes recibidos (se reciban más o menos mensajes que un umbral en un periodo determinado) o por agregación de un campo. Es decir, que su valor agregado sea más alto o bajo que un umbral dado.

Para esta primera condición elegimos el stream DNS y como tipo de condición el valor de un campo (Field Content Alert Condition). Una vez elegidos estos valores nos aparecerá una ventana dónde debemos rellenar una serie de datos, entre los que destacan el campo a evaluar y el valor que debe tener. Además es posible establecer una consulta, de forma que dentro del stream seleccionado se ejecutará dicha consulta y sólo se evaluará la condición sobre su resultado. Podemos ver los datos introducidos en la siguiente imagen:

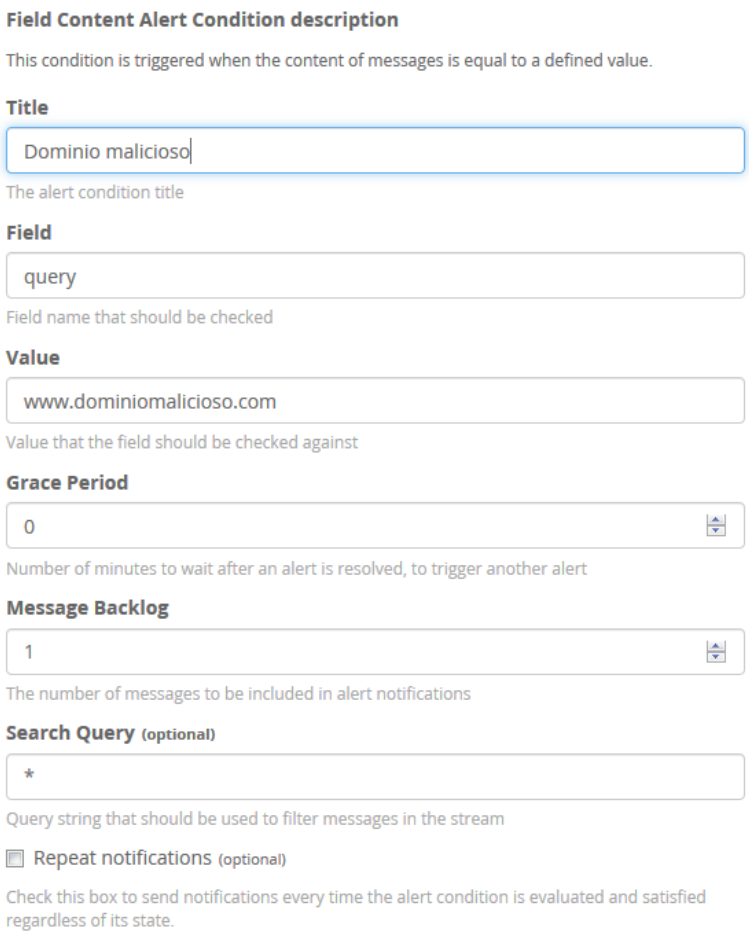


Figura 15: Configuración de la condición “Dominio malicioso”

Para configurar la condición que detectará un número elevado de peticiones DNS se ha seleccionado el stream DNS y el tipo de condición que se usará en este caso será por la cantidad de mensajes recibidos (Message Count Alert Condition).

Una vez elegidos esos datos se nos pedirá configurar la condición de alerta. Entre los datos que se nos piden están la ventana temporal (el periodo de tiempo en el que se tienen que recibir la candad de mensajes especificados), el umbral (número de mensajes recibidos) y si la condición se cumple cuando se reciben más o menos mensajes que el umbral.

En este caso hemos establecido una ventana temporal de 45 minutos y que la condición se cumple cuando se reciben más de cuatro mensajes. De esta forma podemos hacer que la condición se cumpla fácilmente y comprobar su correcto funcionamiento. Al igual que en el caso anterior podemos indicar una consulta para que la condición sólo se evalúe en los resultados de la misma.

En la siguiente imagen podemos ver esta pantalla de configuración:

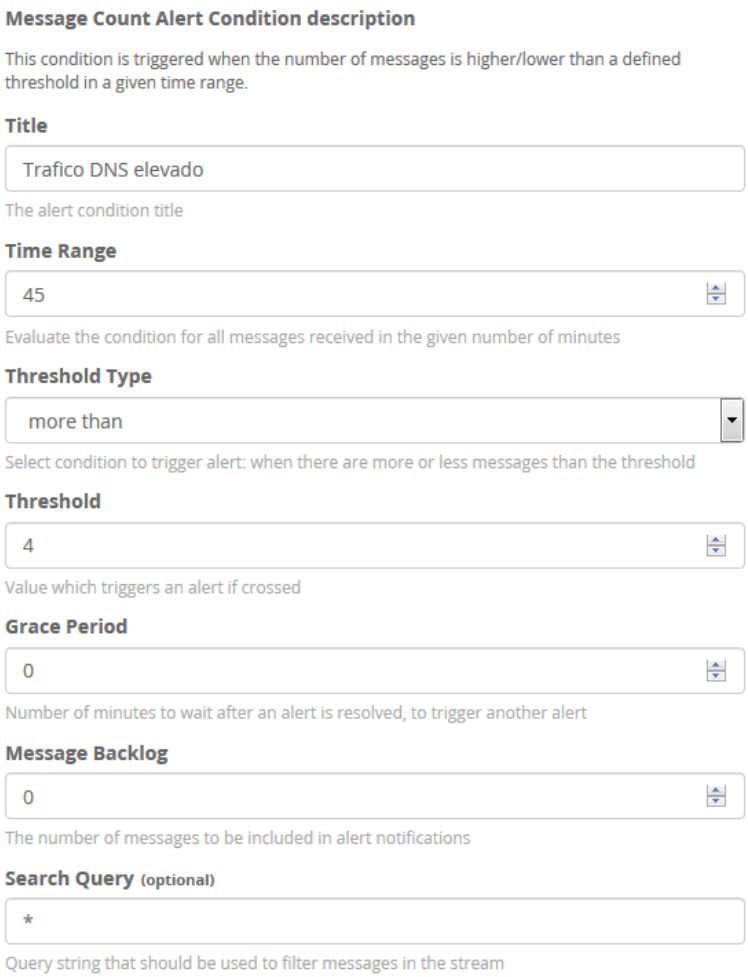


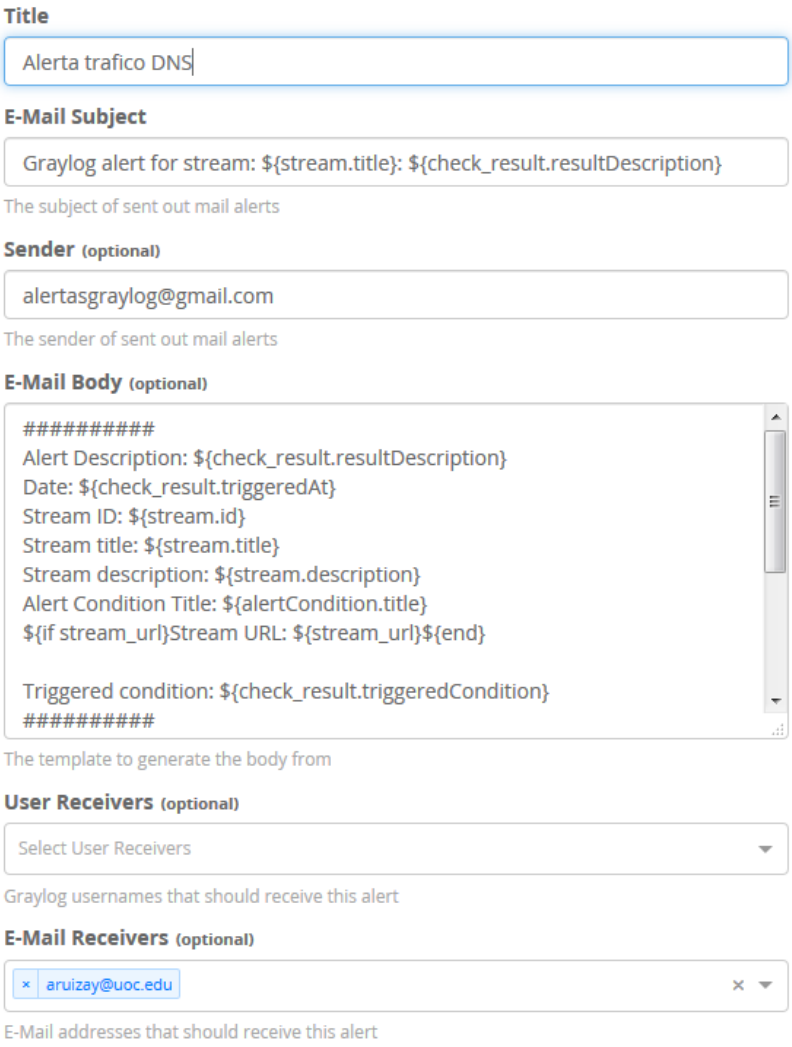
Figura 16: Configuración de la condición “Tráfico DNS elevado”

Una vez hecho esto ya tenemos definidas las condiciones que generarán una alerta. El siguiente paso es configurar su notificación. Para ello vamos al apartado “Notifications”. Creamos una notificación nueva y vemos que en primer lugar se nos piden dos datos:

- El stream en el que se puede activar la alerta que queremos notificar. Al igual que en la definición de las condiciones sólo podemos elegir uno y se notificará cuando se active alguna alerta por cumplirse alguna de las condiciones que tenga configuradas dicho stream.
- El tipo de notificación. Puede ser mediante HTTP o mediante correo electrónico.

Para la notificación que vamos a crear elegimos el stream DNS y que se notifique mediante correo electrónico. Una vez elegidos estos datos se nos pedirá configurar la notificación. Entre los datos que podemos elegir se encuentra el asunto del correo que se enviará, el remitente, la dirección de correo a la que se enviará la notificación y el mensaje que incluirá el correo. Tanto en el asunto como en el mensaje se pueden incluir datos sobre el stream o la condición que activó la alerta a través de variables.

La configuración dada a la notificación es la siguiente:



Como se puede observar el correo se enviará desde una cuenta de gmail creada para este propósito y el destinatario será la cuenta de correo de la UOC. Aunque hay una única notificación que se va a enviar cuando se active cualquiera de las dos condiciones será posible saber que condición se ha cumplido al incluirse en el cuerpo del correo.

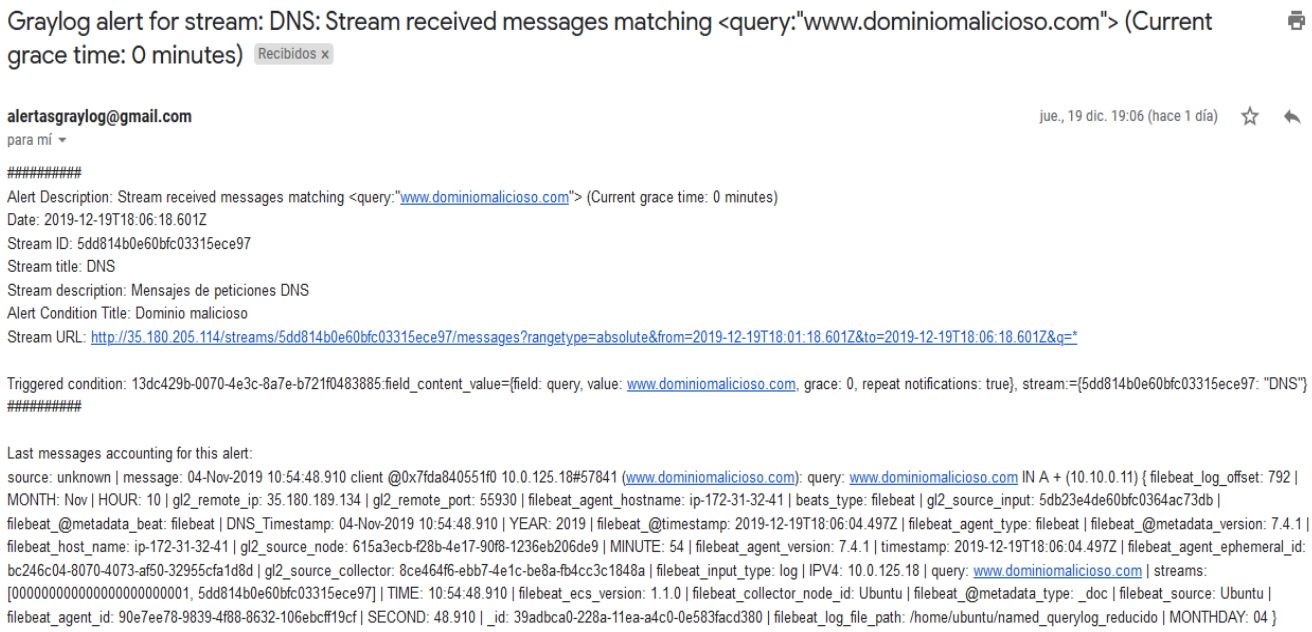
Con esto ya están correctamente configuradas las alertas. Pero dado que la notificación se hace por correo debemos configurar el fichero /etc/graylog/server/server.conf para, entre otras cosas, indicar que SMTP de salida debe utilizar. En este caso utilizaremos uno de gmail, utilizando una cuenta de correo creada expresamente para enviar las notificaciones. Concretamente añadimos en dicho fichero la siguiente información:

transport_email_enabled = true
transport_email_hostname = smtp.gmail.com
transport_email_port = 465
transport_email_use_auth = true
transport_email_use_tls = true
transport_email_auth_username = alertasgraylog@gmail.com (la dirección de la cuenta de correo creada para enviar las notificaciones)
transport_email_auth_password = contraseña de la cuenta anterior
transport_email_subject_prefix = [graylog]
transport_email_from_email = alertasgraylog@gmail.com

Una vez configurado y guardado dicho fichero ya tenemos todo listo para el envío de las notificaciones de alertas por correo.

Para activar las alertas y comprobar su funcionamiento volvemos a enviar una pequeña muestra se los eventos que contiene el log DNS. Dado que para la alerta por cantidad de consultas realizadas se activa cuando se reciben más de 4 mensajes en 45 minutos esta muestra puede ser pequeña. Además, editamos el fichero de logs para que la consulta de alguna de ellas sea la dirección www.dominiomalicioso.com y que así se cumpla la condición de que se este consultando dicho dominio. Una vez enviados a graylog dichos eventos vemos como en el correo de la UOC recibimos 2 notificaciones, una por cada una de las condiciones.

A continuación podemos ver dichos correos:



Graylog alert for stream: DNS: Stream received messages matching <query:"www.dominiomalicioso.com"> (Current grace time: 0 minutes) Recibidos ×

alertasgraylog@gmail.com jue., 19 dic. 19:06 (hace 1 día)
para mí

##########
Alert Description: Stream received messages matching <query:"www.dominiomalicioso.com"> (Current grace time: 0 minutes)
Date: 2019-12-19T18:06:18.601Z
Stream ID: 5dd814b0e60bfc03315ece97
Stream title: DNS
Stream description: Mensajes de peticiones DNS
Alert Condition Title: Dominio malicioso
Stream URL: http://35.180.205.114/streams/5dd814b0e60bfc03315ece97/messages?rangetype=absolute&from=2019-12-19T18:01:18.601Z&to=2019-12-19T18:06:18.601Z&q=*

Triggered condition: 13dc429b-0070-4e3c-8a7e-b721f0483885:field_content_value={field: query, value: www.dominiomalicioso.com, grace: 0, repeat notifications: true}, stream:={5dd814b0e60bfc03315ece97: "DNS"}
##########

Last messages accounting for this alert:
source: unknown | message: 04-Nov-2019 10:54:48.910 client @0x7fda840551f0 10.0.125.18#57841 (www.dominiomalicioso.com): query: www.dominiomalicioso.com IN A + (10.10.0.11) { filebeat_log_offset: 792 | MONTH: Nov | HOUR: 10 | gl2_remote_ip: 35.180.189.134 | gl2_remote_port: 55930 | filebeat_agent_hostname: ip-172-31-32-41 | beats_type: filebeat | gl2_source_input: 5db23e4de60bfc0364ac73db | filebeat_@metadata_beat: filebeat | DNS_Timestamp: 04-Nov-2019 10:54:48.910 | YEAR: 2019 | filebeat_@timestamp: 2019-12-19T18:06:04.497Z | filebeat_agent_type: filebeat | filebeat_@metadata_version: 7.4.1 | filebeat_host_name: ip-172-31-32-41 | gl2_source_node: 615a3ecb-f28b-4e17-90f8-1236eb206de9 | MINUTE: 54 | filebeat_agent_version: 7.4.1 | timestamp: 2019-12-19T18:06:04.497Z | filebeat_agent_ephemeral_id: bc246c04-8070-4073-af50-32955cfa1d8d | gl2_source_collector: 8ce464f6-ebb7-4e1c-be8a-fb4cc3c1848a | filebeat_input_type: log | IPV4: 10.0.125.18 | query: www.dominiomalicioso.com | streams: [000000000000000000000001, 5dd814b0e60bfc03315ece97] | TIME: 10:54:48.910 | filebeat_ecs_version: 1.1.0 | filebeat_collector_node_id: Ubuntu | filebeat_@metadata_type: _doc | filebeat_source: Ubuntu | filebeat_agent_id: 90e7ee78-9839-4f88-8632-106ebcff19cf | SECOND: 48.910 | _id: 39adbca0-228a-11ea-a4c0-0e583facd380 | filebeat_log_file_path: /home/ubuntu/named_querylog_reducido | MONTHDAY: 04 }

Figura 17: Notificación de consulta DNS sobre un dominio malicioso

Graylog alert for stream: DNS: Stream had 7 messages in the last 45 minutes with trigger condition more than 4 messages. (Current grace time: 0 minutes) Recibidos ×

alertasgraylog@gmail.com jue., 19 dic. 19:06 (hace 1 día)
para mí

##########
Alert Description: Stream had 7 messages in the last 45 minutes with trigger condition more than 4 messages. (Current grace time: 0 minutes)
Date: 2019-12-19T18:06:17.363Z
Stream ID: 5dd814b0e60bfc03315ece97
Stream title: DNS
Stream description: Mensajes de peticiones DNS
Alert Condition Title: Trafico DNS elevado
Stream URL: http://35.180.205.114/streams/5dd814b0e60bfc03315ece97/messages?rangetype=absolute&from=2019-12-19T17:21:17.363Z&to=2019-12-19T18:06:17.363Z&q=*

Triggered condition: 02a6ca9c-a0b1-46d7-abe6-43433b4fa2dc:message_count={time: 45, threshold_type: more, threshold: 4, grace: 0, repeat notifications: true}, stream:={5dd814b0e60bfc03315ece97: "DNS"}
##########

Figura 18: Notificación de tráfico DNS elevado

3.4. Caso práctico 3. Análisis de logs de un WAF

En este apartado vamos a analizar los logs de un WAF, concretamente de un Fortiweb de Fortinet. Como introducción, diremos que un WAF es un firewall específico para aplicaciones web, el cual es capaz de filtrar el tráfico entre un navegador web y el servidor web en ambos sentidos. Por lo tanto, a través de los eventos que genera podremos saber que tipo de ataques ha recibido nuestra web, desde que direcciones IP y como que acción ha llevado a cabo el WAF.

Para este caso práctico analizaremos dos tipos de logs: por un lado los relativos a ataques y por otro los relativos a eventos del sistema que tienen lugar en el WAF.

3.4.1. Desarrollo del caso práctico 3

Para llevar a cabo este caso práctico utilizaremos como ya se ha indicado logs generados por un Fortiweb durante 30 días proporcionados por el tutor de este trabajo, los cuales enviaremos a graylog utilizando filebeat mediante Sidecar.

El formato de estos logs es bastante complejo ya que está formado por multitud de pares de la forma clave=valor. A continuación indicaremos los campos que consideramos más relevantes. La lista completa se puede consultar en la “Log reference” de Fortiweb, mencionada en la bibliografía.

Los siguientes campos están presentes tanto en los mensajes del log de ataques como en el de eventos:

- log_id: valor numérico que identifica la causa por la que se ha creado el mensaje. Todos los eventos creados por la misma causa compartirán el identificador
- msg_id: valor numérico que identifica cada mensaje de forma única.
- type: indica la categória a la que pertenece el mensaje y en que fichero de logs se ha almacenado. Este campo puede tomar el valor “attack”, “traffic” o “event”.
- main_type: tipo principal al que pertenece el evento
- pri: nivel de gravedad del evento que generó el mensaje. Puede ser “Information”, “Notification”, “Warning”, “Error”, “Critical”, “Alert” o “Emergency”.
- action: la acción llevada a cabo.
- msg: este campo incluye una descripción del motivo por el que se ha creado el mensaje en el fichero de logs.

Los siguientes campos son exclusivos de los mensajes del log de ataques:

- sub_type: indica el subtipo al que pertenece el ataque dentro del tipo principal indicado por main_type
- severity_level: nivel de gravedad configurado en la regla que provocó la generación del log.
- policy: el nombre de la política que ha causado el mensaje.
- src: la dirección IP origen del tráfico.
- dst: la dirección IP destino del tráfico
- http_url: la dirección URL que incluye la petición HTTP en su cabecera.
- http_host: host que contiene la petición HTTP en su cabecera.
- signature_subclass: subclase de la firma.
- signature_id: identificador de la firma que ha creado el mensaje.
- srccountry: país orinen del tráfico.

- threat_level: nivel asignado a la amenaza detectada.

Los siguientes campos son exclusivos de los mensajes del log de eventos:

- subtype: indica el subtipo al que pertenece el evento dentro del tipo indicado por main_type.
- status: el resultado de la acción llevada a cabo.

Como podemos ver, aunque los logs de ataques y eventos tienen campos en común, tienen otros específicos. Y dado que la información que aportan es totalmente distinta en cada caso (ataques por un lado y eventos del sistema por otro), lo ideal es poder tratarlos por separado. Para ello nos vamos a crear dos streams, uno para los logs de ataques y otro para los de eventos. En ambos casos para que los mensajes sean enrutados a ellos estos tendrán que cumplir con una única regla. Esta regla será que cumpla una expresión regular, y para que los mensajes de ataques puedan ser distinguidos de los de logs utilizaremos alguno de sus campos exclusivos para hacer la expresión regular.

En el caso de los logs de ataques creamos un stream llamado “Waf Attacks” cuya regla será que cumpla con la expresión regular “.http_method.” el cual es un campo exclusivo de los logs de ataques aunque no se ha considerado de los más relevantes, por eso no se ha mencionado antes.

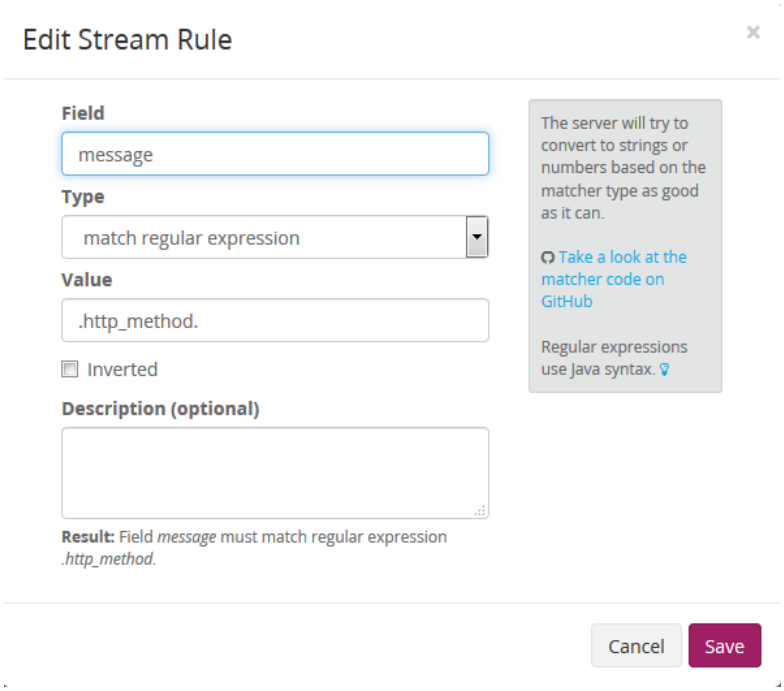


Figura 19: Regla del stream “Waf Attacks”

En cuanto a los logs de eventos creamos un stream llamado “Waf Events” cuya regla será que cumpla con la expresión regular “.user=.” ya que el campo user es exclusivo de los logs de eventos.

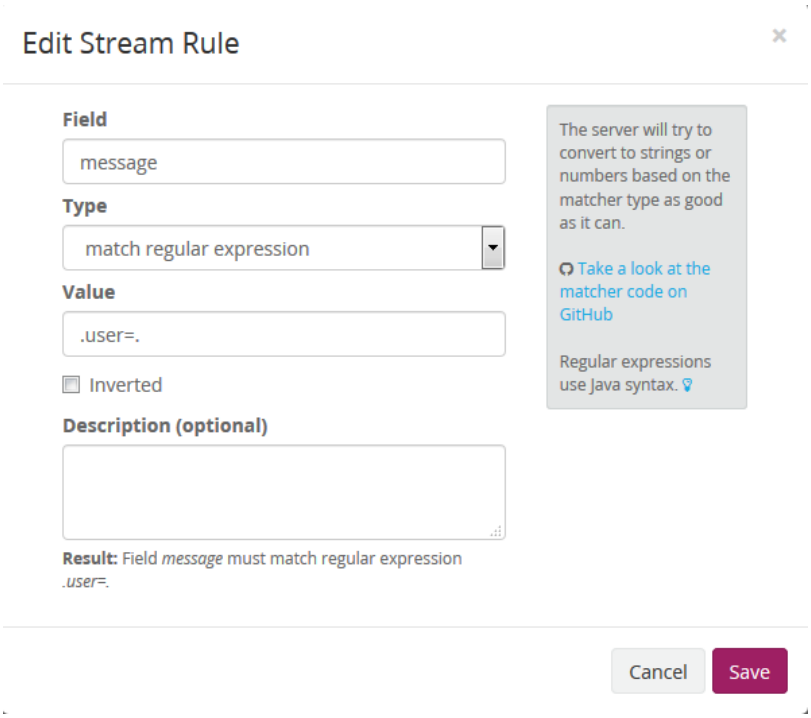


Figura 20: Regla del stream “Waf Events”

Una vez hemos configurado los streams para tener los mensajes correctamente clasificados el siguiente paso es extraer la información de ellos. Para ello habrá que configurar un extractor.

Como ya se ha mencionado, el formato que tienen estos mensajes de log es muy complejo ya que tienen multitud de campos. Por ello, intentar extraer toda su información a través de expresiones regulares y patrones Grok puede ser una ardua tarea. Por suerte, a pesar de tener multitud de campos todos son de la forma clave=valor y graylog tiene un extractor para extraer este tipo de información. Una vez este configurado correctamente el extractor, lo que hará graylog es copiar información del mensaje, creando una variable para cada clave y asignándole automáticamente su valor.

Para ello, creamos un extractor de tipo “Copy Input” en el Input por el que llegarán los mensajes (en este caso el Input “Beats” que ya tenemos. Una vez en la pantalla de configuración tenemos que indicar los datos de la siguiente imagen:

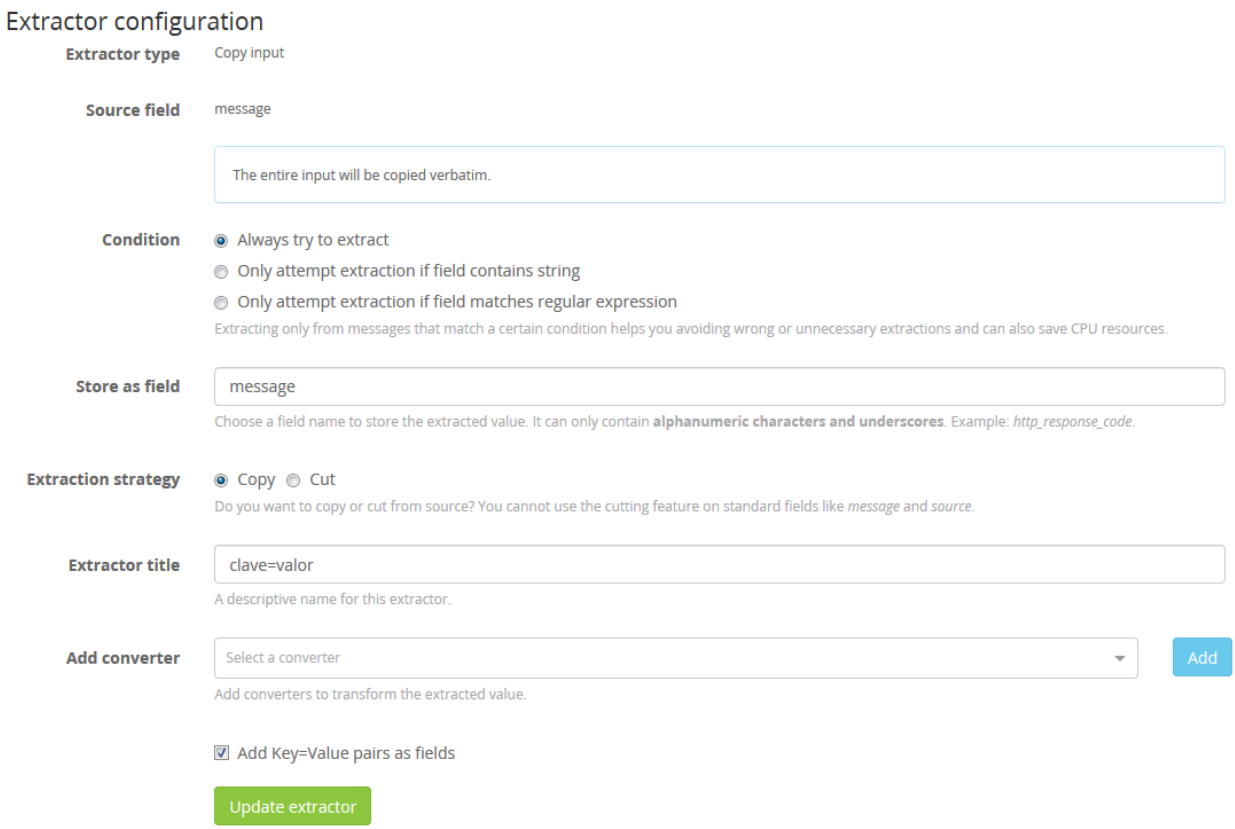


Figura 21: Extractor “clave=valor”

Hay que destacar que abajo, dónde se indica “Add converter” hay que seleccionar el conversor “Key=Value pairs as field” y añadirlo.

Una vez creado el extractor este funcionará para cualquier mensaje que contenga información de la forma clave=valor y llegue a través de este Input. Por lo tanto, este extractor nos servirá tanto para los mensajes de ataques como los de eventos, así como para mensajes futuros.

3.4.2. Resultado del análisis de logs de un WAF

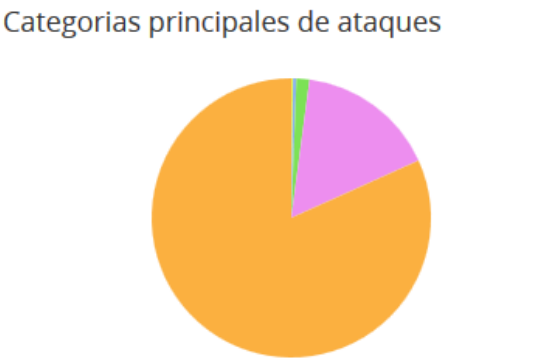
Una vez ya hemos configurado los streams y los extractores necesarios podemos recibir los mensajes y tenerlos correctamente clasificados y además obtener información de ellos que podemos usar para hacer búsquedas de interés.

En este apartado toda la información que saquemos será de los últimos 30 días.

Empezaremos por analizar los logs de ataques.

En primer lugar, vamos a crear un cuadro de mando en “dashboards” llamado WAF attacks al que le iremos añadiendo diversa información. Todos los resultados gráficos que se van a mostrar en este apartado están incluidos en dicho cuadro de mando.

Lo primero que nos puede interesar saber es que tipo de ataques estamos recibiendo. Esto podemos verlo a través de 2 variables: log_id o main_type. Dado que la primera es un identificador numérico optaremos por la segunda opción, ya que en caso de utilizar el log_id solamente veriamos un número y tendríamos que consultar la documentación del WAF para ver con que main_type se corresponde. El resultado es el siguiente:



Value	%	Count
Top 5 values		
Signature Detection	81.74%	376,636
HTTP Protocol Constraints	16.16%	74,471
DoS Protection	1.50%	6,929
IP Reputation	0.39%	1,810
Brute Force Login	0.20%	910

Figura 22: categorías principales de ataques recibidos

Una vez hemos obtenido esta información lo siguiente que nos interesa es saber cuales son los subtipos de ataques recibidos puesto que estos son tipos de ataque concretos. Esta información la obtenemos a través de la variable sub_type. Al igual que en caso anterior podemos hacer que graylog nos muestre los valores de esta variable de forma gráfica:

Value	%	Count
Top 5 values		
Bad Robot	61.74%	284,456
HTTP Duplicated Parameter Name	6.91%	31,818
URL Parameter Number Overflow	6.68%	30,795
SQL Injection	6.50%	29,935
Generic Attacks	5.15%	23,744
Others		
Information Disclosure	3.91%	17,993
SQL Injection (Syntax Based Detection)	1.84%	8,479
Cross Site Scripting	1.78%	8,203

Figura 23: Subtipos de ataques recibidos

Ahora bien, el gráfico anterior nos muestra los subtipos de ataques recibidos sin tener en cuenta a que tipo principal pertenecen. Esto es útil, ya que como se puede ver en la imagen podemos ver de forma rápida los tipos concretos de ataque que estamos recibiendo. Pero podría interesarnos conocer esta información ordenada por tipo principal de ataque. Esto podemos hacerlo realizando la búsqueda dentro del stream de la forma: main_type:"tipo principal de ataque". En la siguiente imagen podemos ver los resultados de estas búsquedas:

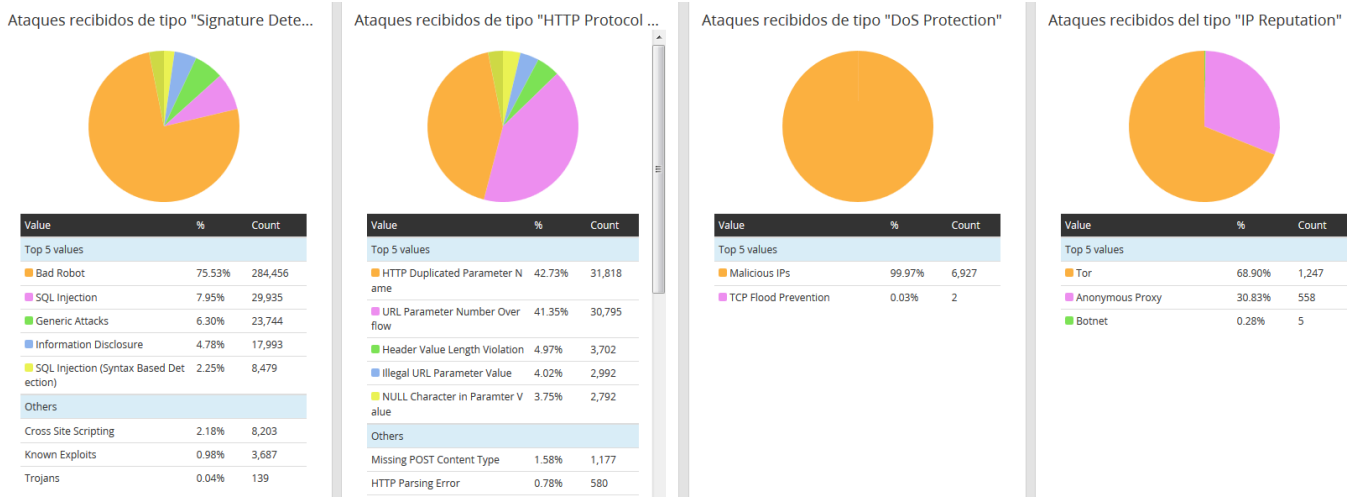


Figura 24: Subtipos de ataque recibidos para cada tipo principal

A continuación vamos a ver como de graves son los ataques recibidos, mediante la variables threat_level:

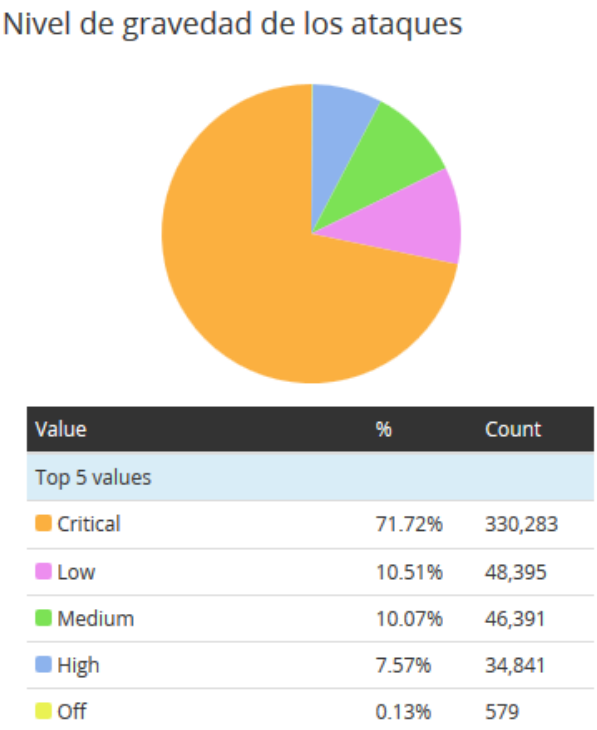


Figura 25: Nivel de gravedad de los ataques

Una vez tenemos información sobre los tipos de ataques que estamos recibiendo vamos a pasar a ver en primer lugar desde dónde se están recibiendo los ataques, tanto mediante la dirección IP a través de la variable src:

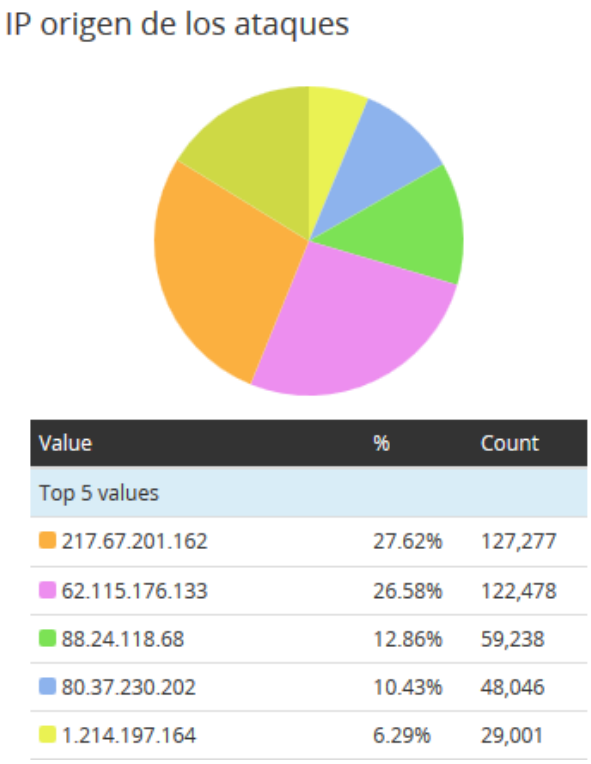


Figura 26: IP origen de los ataques

Otro dato que nos puede interesar es saber desde que países se están recibiendo los ataques, lo cual podemos saber a través de la variable srccountry. Este dato es de interés ya que en un firewall se pueden poner reglas de forma que a partir de una numéro determinado de sesiones desde la misma IP esta quede baneada durante un periodo determinado. Si el firewall tiene capacidad de geolocalizar las IP podemos situar este umbral más bajo para los países desde los que más ataques se reciban.

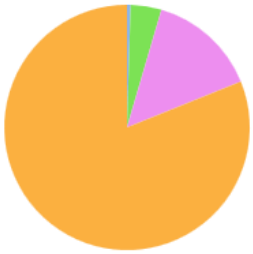
Paises origen de los ataques

Value	%	Count
Top 5 values		
Poland	27.66%	127,350
Europe	26.60%	122,505
Spain	24.72%	113,824
Republic of Korea	6.31%	29,058
Reserved	5.90%	27,172
Others		
United States	2.42%	11,165
Netherlands	1.18%	5,417
Panama	0.99%	4,569
Czech Republic	0.99%	4,554

Figura 27: Países origen de los ataques

Lo siguiente que deberíamos saber es el destino de los ataques, a través de la variable dst:

IP destino de los ataques



Value	%	Count
Top 5 values		
10.134.19.132	81.11%	373,736
10.134.19.110	14.35%	66,138
192.168.3.132	4.06%	18,716
192.168.3.64	0.47%	2,166

Figura 28: IP destino de los ataques

También mirados por qué motivos se han generado los mensajes del log, a través de la variable msg:

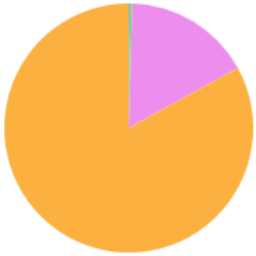
Motivos de creacion de los mensajes

Value	%	Count
Top 5 values		
HTTP Header triggered signature ID 110000027 of Signatures policy PRD-LIFERAY	53.73%	247,448
[policy_name=PRD-PHP] : Duplicated Parameter Name	6.44%	29,648
HTTP Header triggered signature ID 110000022 of Signatures policy PRD-APPS_PLATINO	3.01%	13,845
[policy_name=PRD-PHP] : Too Many Parameters in Request: (The number of url parameters in request (22) exceeded the maximum allowed - 16)	2.22%	10,240

Figura 29: Motivos de creación de los mensajes del log

Finalmente, vemos cuales son las acciones que se han llevado a cabo mediante la variable action:

Accion llevada a cabo



Value	%	Count
Top 5 values		
Alert	83.00%	382,406
Alert_Deny	16.54%	76,189
Erase	0.27%	1,251
Period_Block	0.20%	910

Figura 30: Acción llevada a cabo

Toda esta información que hemos mostrado y añadido a un cuadro de mando es información global que nos puede interesar. Pero viendo en cada momento lo que nos muestra podemos realizar búsquedas para ver cosas más concretas. Por ejemplo, hemos visto que los dos tipos de ataque concreto (subtipos) más recibidos son “Bad Robot” y “SQL Injection”.

Dado que los ataques de SQL Injection son muy dañinos queremos saber más información sobre ellos, así que buscamos sub_type:"SQL Injection" y entre los resultados que nos muestra graylog queremos saber que acción se ha llevado a cabo. Para ello vemos los “quick values” de la variable “action”:

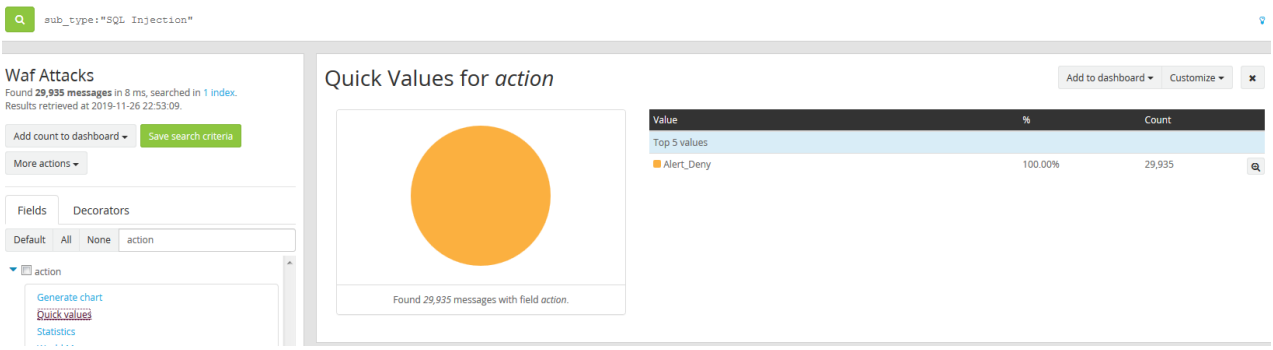


Figura 31: Acciones llevadas a cabo para ataques SQL Injection

Y vemos que en todos los casos la acción llevada a cabo es Alert_Deny. De forma análoga podemos ver de que países están procediendo dichos ataques, mediante los quick values de srccountry:

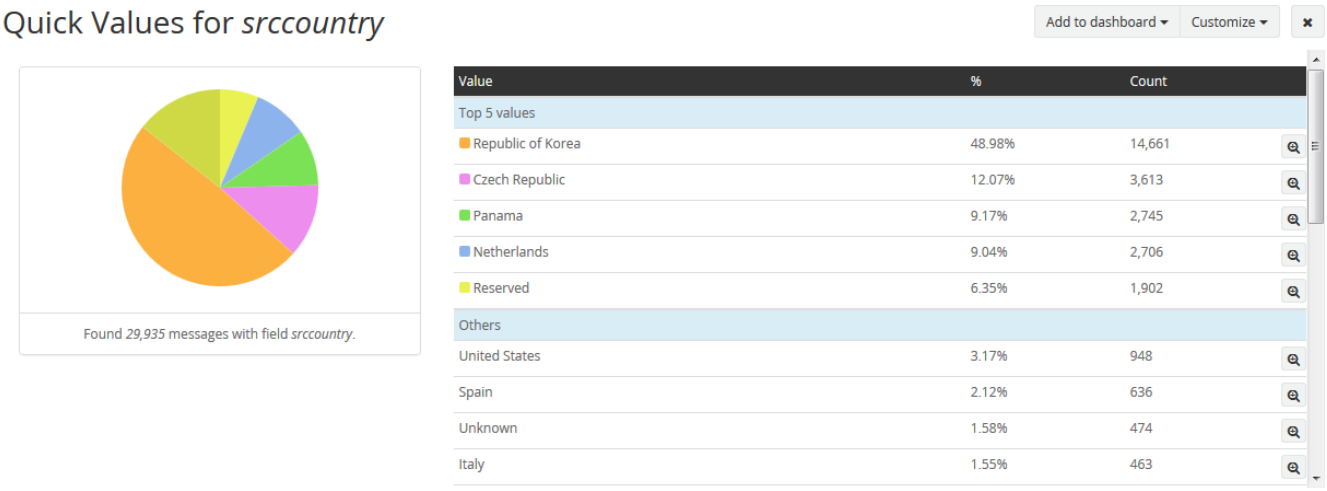


Figura 32: Países origen de los ataques SQL Injection

Como podemos ver, los recibidos desde España son tan sólo el 2,12%. Pero podríamos querer saber que IP son para poder informar a las autoridades de que una serie de IP españolas nos están atacando. Así que hacemos la búsqueda: *sub_type:"SQL Injection" AND srccountry:Spain* y dentro de sus resultados consultamos las direcciones IP mediante los Quick values de src para ver que todos los ataques proceden desde la misma dirección IP:

Figura 33: IP españolas que realizan ataques SQL Injection

Una vez hemos analizado los logs de ataques vamos a pasar a analizar los logs de eventos. Para ello creamos un cuadro de mando dentro de Dashboards llamado WAF Events y entramos en el stream del mismo nombre para obtener información.

En primer lugar vamos a ver que tipos de eventos concretos están teniendo lugar mediante la variable subtype, y cómo vemos todos son eventos del sistema:

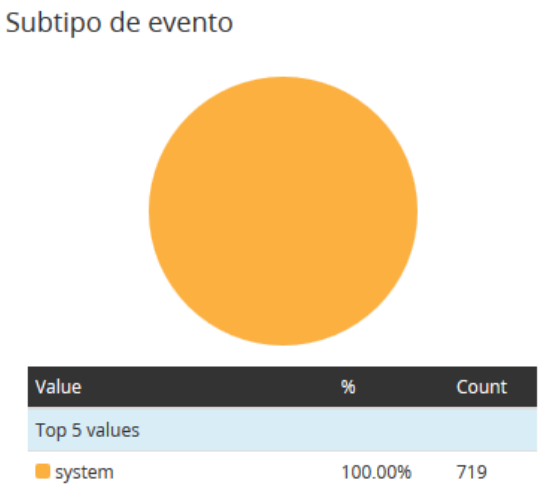
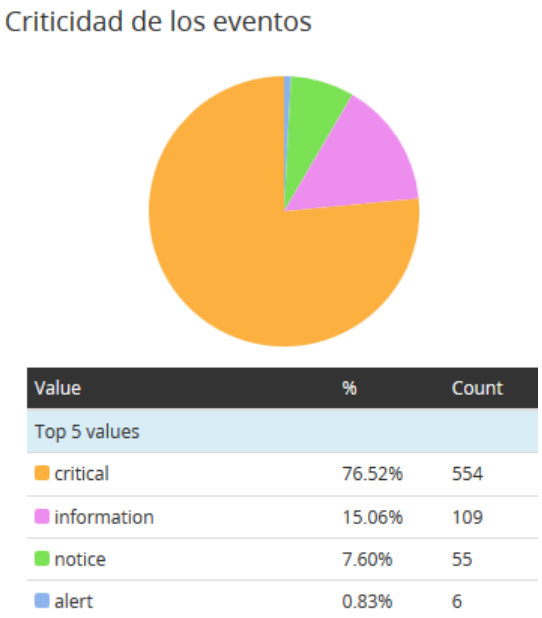


Figura 34: Tipos de evento

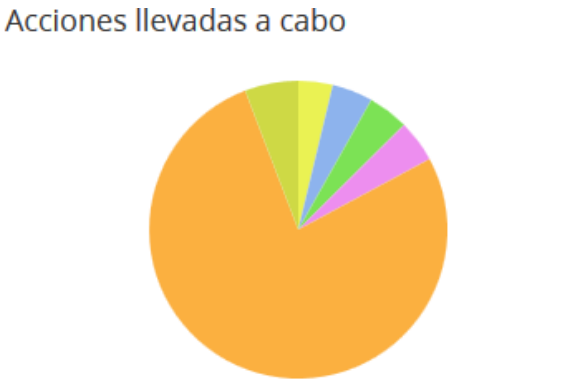
lo siguiente que nos interesa es saber que nivel de criticidad tienen dichos eventos, mediante la variable pri:



Value	%	Count
Top 5 values		
critical	76.52%	554
information	15.06%	109
notice	7.60%	55
alert	0.83%	6

Figura 35: Criticidad de los eventos

También nos interesa saber que acciones se han llevado a cabo mediante la variable “action”:



Value	%	Count
Top 5 values		
update	77.07%	558
browse	4.56%	33
edit	4.42%	32
backup	4.42%	32
login	3.73%	27

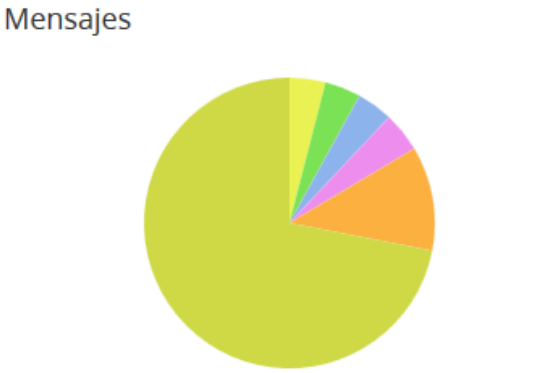
Figura 36: Acciones llevadas a cabo

Y el estado de las mismas mediante la variable “status”:

Value	%	Count
Top 5 values		
success	52.29%	376
failure	47.71%	343

Figura 37: Estado de las acciones

Y finalmente, los motivos por los que se han creado los mensajes en el log, mediante la variable “msg”:



Value	%	Count
Top 5 values		
Failed to establish connection with 173.243.138.80:443 when install essential/anti-virus packages.	11.60%	84
Reseeding successfully from the old method	4.42%	32
Fortiweb FDNI update succeeded	4.01%	29
Fortiweb FSCI update succeeded	4.01%	29
Fortiweb FCNI update succeeded	4.01%	29

Figura 38: Motivos por los que se han creado los mensajes

Al igual que en el caso de los log de ataques esta información es global, y según lo que veamos podemos hacer búsquedas más concretas para ver que está ocurriendo. En este caso por ejemplo, vemos que más del 47% de las acciones han fallado. Así que nos puede interesar saber en primer lugar que tipo de acciones han fallado. Para ello hacemos la búsqueda: *status:failure* y al mostrarnos su resultado consultamos los quick values de la variable “action:

Value	%	Count
Top 5 values		
update	94.46%	324
check-resource	5.25%	18
login	0.29%	1

Figura 39: Acciones que han fallado

Y vemos que la gran mayoría de los casos son actualizaciones fallidas. Así que queremos saber los principales motivos. Para ello realizamos la búsqueda: *status:failure AND action:update* y consultamos los quick values de la variable msg:

Value	%	Count
Top 5 values		
Failed to establish connection with 173.243.138.80:443 when install essential/anti-virus packages.	25.93%	84
Fortiweb HCDB is unauthorized	8.95%	29
Failed to establish connection with 173.243.138.68:443 when install essential/anti-virus packages.	6.17%	20
Failed to establish connection with 173.243.138.66:443 when install essential/anti-virus packages.	5.56%	18
Failed to establish connection with 173.243.138.70:443 when install essential/anti-virus packages.	5.56%	18

Figura 40: Principales motivos por los que han fallado actualizaciones

3.5. Caso práctico 4. Análisis de eventos de un Firewall

En este apartado vamos a analizar los eventos de un firewall de PaloAlto proporcionados por el tutor de este trabajo.

Es interesante analizar los eventos de un firewall ya que de esta manera podemos averiguar que tipo de tráfico estamos teniendo: a que aplicaciones corresponde el tráfico, cuál es su origen y destino, etc.

Además podemos averiguar que tráfico se está permitiendo o denegando. A través del análisis de este tráfico podríamos darnos cuenta de situaciones cómo pueden ser:

- Que uno de nuestros equipos este estableciendo un gran número de sesiones hacia múltiples IP destino en algún puerto determinado (por ejemplo el 22, SSH) para intentar algún tipo de ataque. Esto podría indicar que el equipo se encuentre infectado por malware o que haya alguna persona utilizándolo con fines ilícitos.
- Que observemos una gran cantidad de tráfico que no se deniega hacia puertos en los que suele correr alguna aplicación con vulnerabilidades conocidas con el objetivo de explotarlas. Esta situación podría hacer que nos planteemos filtrar el tráfico hacia dichos puertos.
- Uso excesivo de red por parte de alguna aplicación. En este caso tendríamos que averiguar a que se debe esta situación.

3.5.1. Desarrollo del caso práctico 4

Para realizar este caso práctico, como se ha mencionado se han utilizado los eventos de un firewall de PaloAlto proporcionados por el tutor de este trabajo. Dichos eventos están en formato LEEF y se han enviado a graylog mediante Beats utilizando Sidecar.

En primer lugar, al igual que hicimos en los casos anteriores, vamos a enrutar los mensajes del firewall en un stream (llamado Firewall). De esta forma será más sencillo realizar búsquedas únicamente sobre eventos del firewall al tenerlos separados de otros tipos de eventos.

Si vemos los eventos del firewall veremos que todos incluyen la marca del fabricante (Palo Alto Networks). Dado que no tenemos ningún otro tipo de evento que incluya dicho texto haremos que se enruten al stream Firewall todos los eventos que incluyan la expresión regular “.Palo Alto Networks.”

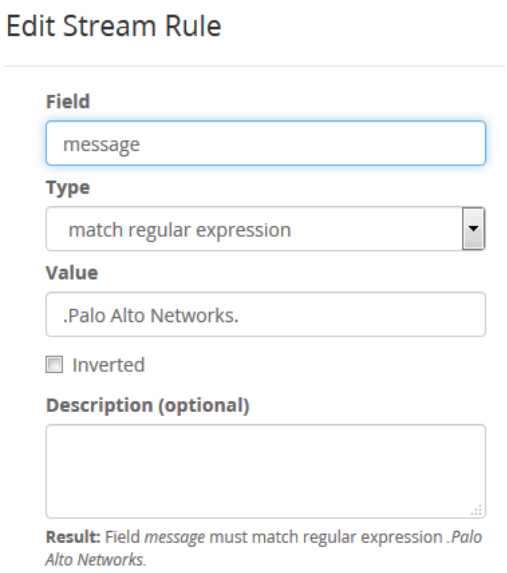


Figura 41: regla del stream “Firewall”

Cada evento del log tiene múltiples campos de los cuáles tan sólo utilizaremos los de mayor interés. Concretamente, utilizaremos los siguientes campos:

- action: indica si se ha permitido o rechazado el paquete.
- src y dst: indican la dirección IP origen y destino respectivamente.
- srcPort y dstPort: indican el puerto origen y destino respectivamente.
- RuleName: indica el nombre de la regla que se ha aplicado.
- Application: indica a que aplicación pertenece el tráfico. No siempre se va a determinar esta aplicación.
- SessionID: indica el identificador de la sesión.
- proto: indica el protocolo utilizado; tcp o udp.
- SessionEndReason: indica el motivo por el que ha finalizado la sesión.

Algunos de estos datos los lee graylog al recibir los mensajes. El problema es que no los lee de forma individual, si no que en un mismo campo almacena varios datos, cómo por ejemplo src y dst. Por ello vamos a optar por crear extractores de forma que tengamos un campo para cada dato.

Para extraer la dirección IP origen y destino crearemos dos extractores llamados srcip y dstip que almacenarán dichas IP en un campo del mismo nombre. para ello nos basaremos en el patrón Grok existente en graylog para extraer una IPv4. Aunque anteriormente ya habíamos creado este extractor en el caso práctico del análisis de eventos DNS no nos es posible usarlo, ya que ese extractor extraerá únicamente la primera dirección IP que aparezca en el evento y queremos extraer las dos. Por ello utilizaremos el siguiente patrón Grok para almacenar la dirección IP origen:

src=%{IPV4:srcip}

dónde IPV4 es el patrón Grok ya existente para extraer una dirección IP. De forma análoga el patrón para extraer la dirección IP destino es el siguiente:

dst=%{IPV4:dstip}

De esta forma almacenamos el valor de src y dst en los campos srcip y dstip respectivamente.

Para extraer la acción y el protocolo, dado que son campos que tan solo pueden contener uno de dos valores (allow o deny la acción y tcp o udp el protocolo) utilizaremos una expresión regular en la que se indicará con una alternativa que el valor a extraer es uno de esos dos. Llamaremos a estos extractores action y proto y se almacenará su valor en un campo del mismo nombre. Por tanto las expresiones regulares quedan así:

action=(allow|deny)
proto=(tcp|udp)

Para el resto de extractores utilizaremos expresiones regulares. Para todas ellas utilizaremos en principio una expresión regular de la forma “campo=(.*)” . Pero con esta expresión regular no extraeremos sólo el valor del campo que nos interesa, si no también todo lo que venga a continuación en el mensaje que contiene el evento. Para evitar esto y extraer sólo el valor del campo que nos interesa la solución utilizada es, dado que conocemos el formato del log el cual nunca varía, añadir el nombre del campo que venga a continuación al final de la expresión regular. De esta forma, tan sólo se extraerá el valor que nos interesa. Además, para aquellos campos que tienen un valor numérico añadimos un conversor a valor numérico. El resultado del extractor se almacenará en un campo del mismo nombre que el campo original, salvo en el caso de SessionEndReason, que se almacenará en un campo llamado SessionEnd. El motivo es que graylog extrae dicha información, junto a otra adicional en un campo llamado SessionEndReason, por lo que no podemos utilizar dicho nombre.

De esta forma las expresiones regulares quedan así.

srcport: srcPort=(.+)\|dstPort
dstport: dstPort=(.+)\|ssrcPostNatPort
RuleName: RuleName=(.*)\|usrName
Application: Application=.(.*)\|VirtualSystem
SessionID: SessionID=(.+)\|RepeatCount
SessionEnd: SessionEndReason(.*)\|DeviceGroupHierarchyL1

A modo de ejemplo en la siguiente imagen se puede ver la pantalla de configuración del extractor srcport al cual se ha añadido el conversor a valor numérico:

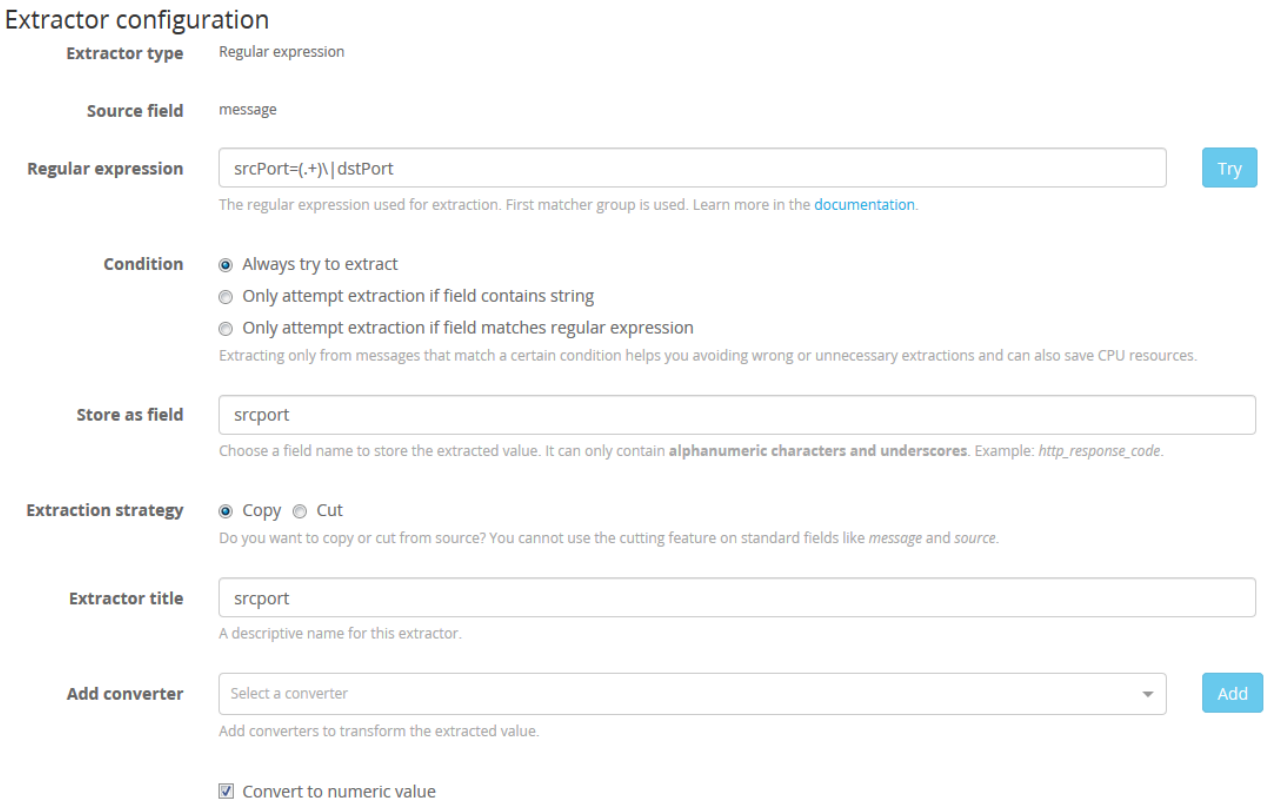


Figura 42: Extractor srcport

3.5.2. Resultados del análisis de eventos de un firewall

Una configurados correctamente el stream y los extractores necesarios podemos recibir los eventos del firewall y poder obtener de ellos la información que necesitemos. Toda la información que vamos a ir mostrando en este apartado la añadiremos a un cuadro de mando (Dashboard) llamado firewall. En primer lugar vamos a añadir información que nos puede interesar tener disponible en todo momento. Las búsquedas realizadas serán sobre la totalidad de los eventos enviados a graylog, que se corresponden con un día, lo que supone más de 1600000 eventos.

En primer lugar vamos a ver a que aplicaciones pertenece el tráfico que pasa a través del firewall. Además vamos a hacer esta consulta tanto para el tráfico permitido como denegado. Empezamos con el tráfico permitido, para ello entramos al stream Firewall y buscamos en todos los mensajes añadiendo en la búsqueda el filtro action:allow. Una vez obtenido el resultado y añadida la información en forma de gráfica al cuadro de mando, repetimos la búsqueda para el tráfico denegado añadiendo al filtro action:deny. Al obtener el resultado y ver sus “quick values” vemos que el primero es “not-applicable”. Esto no nos dice mucho, así que vamos a excluirlo del resultado para que tán solo aparezcan

aplicaciones. Para ello realizamos la búsqueda “action:deny –Application:not-applicable”. Una vez obtenido el resultado añadimos la gráfica del mismo al cuadro de mando. En la siguiente imagen se pueden ver los resultados:

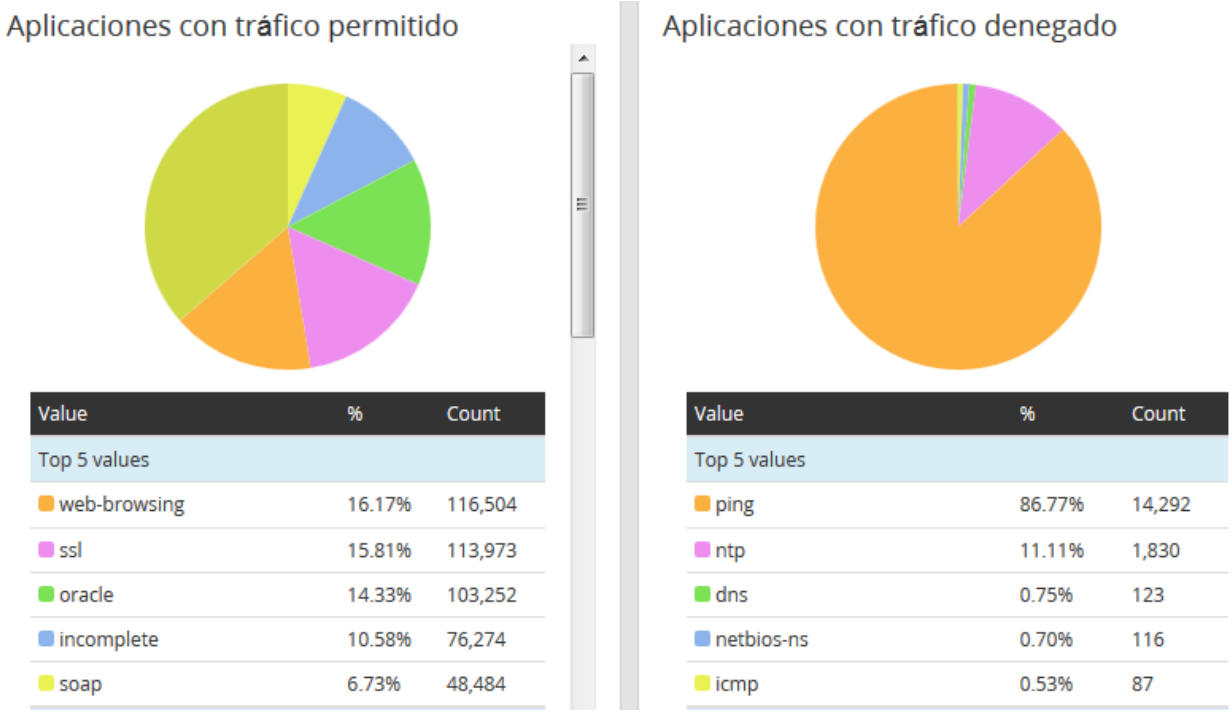


Figura 43: Resultados de tráfico de aplicaciones permitido y denegado

A continuación vamos a ver que reglas se están aplicando mediante los “quick values” del campo “RuleName”:

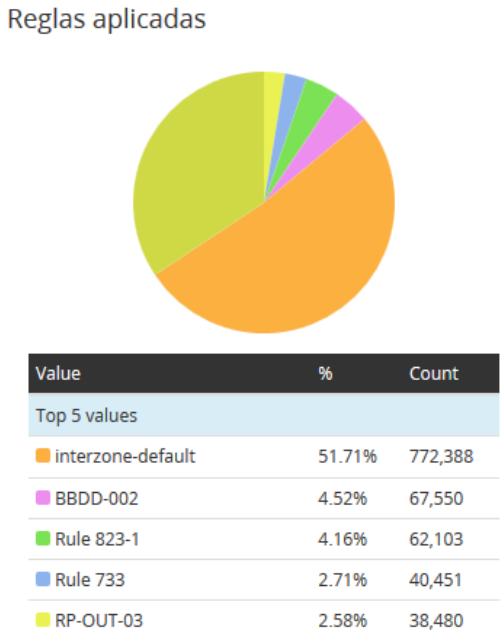


Figura 44: Reglas aplicadas

Lo siguiente que vamos a ver son los puertos de destino, tanto cuando el protocolo utilizado es tcp como cuando es udp. Además vamos a hacer la consulta tanto para tráfico permitido como denegado. Para ello haremos las consultas utilizando los campos action y proto. Concretamente, las búsquedas realizadas son las siguientes:

action: allow AND proto:tcp
action: deny AND proto:tcp
action: allow AND proto:udp
action: deny AND proto:udp

Y el resultado de las mismas puede verse en los siguientes gráficos obtenidos mediante los “quick values” del campo dstport:

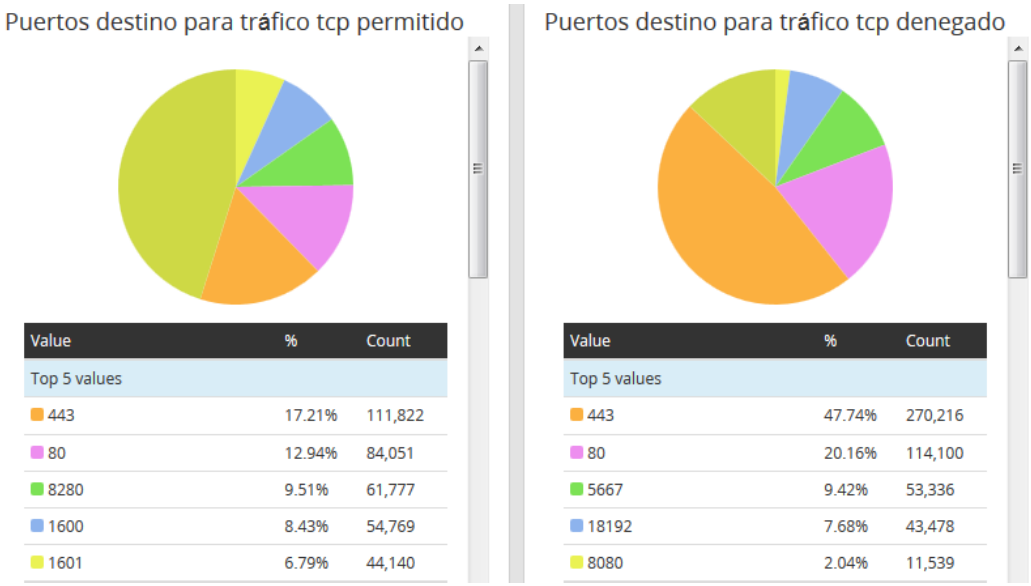


Figura 45: tráfico tcp permitido y denegado

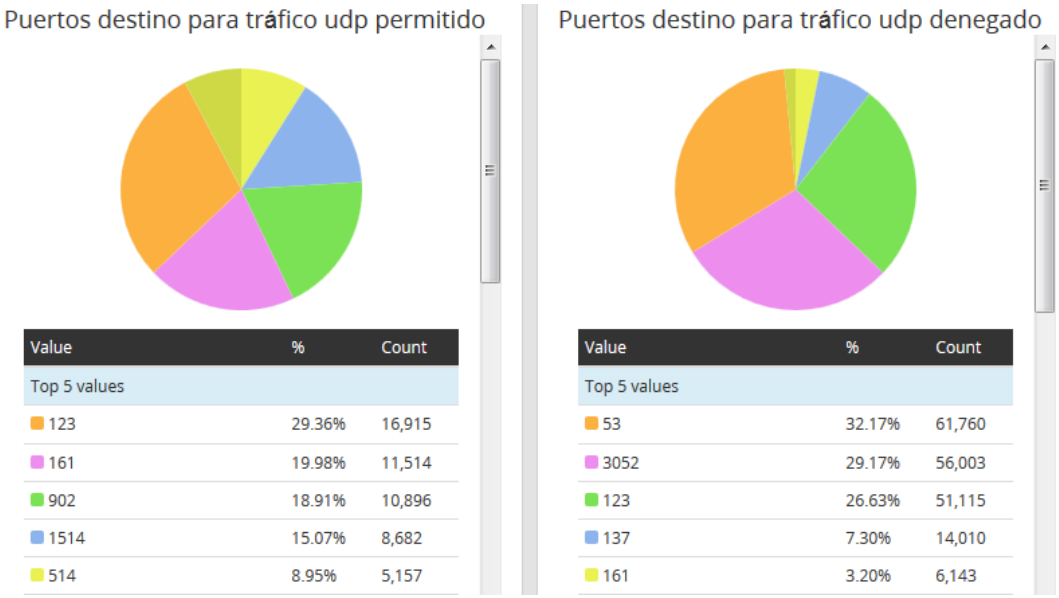


Figura 46: tráfico udp permitido y denegado

A continuación vemos el motivo de finalización de las sesiones, obtenido mediante los “quick values” del campo sessionEnd:

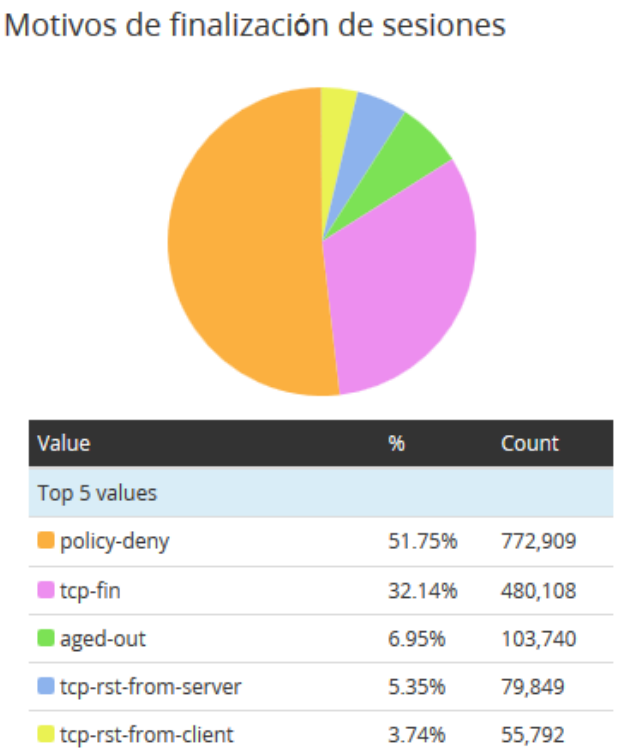


Figura 47: Motivos de finalización de sesiones

Por último, vamos a agregar al cuadro de mando los ranking de direcciones IP origen y destino, tanto para tráfico permitido como denegado.

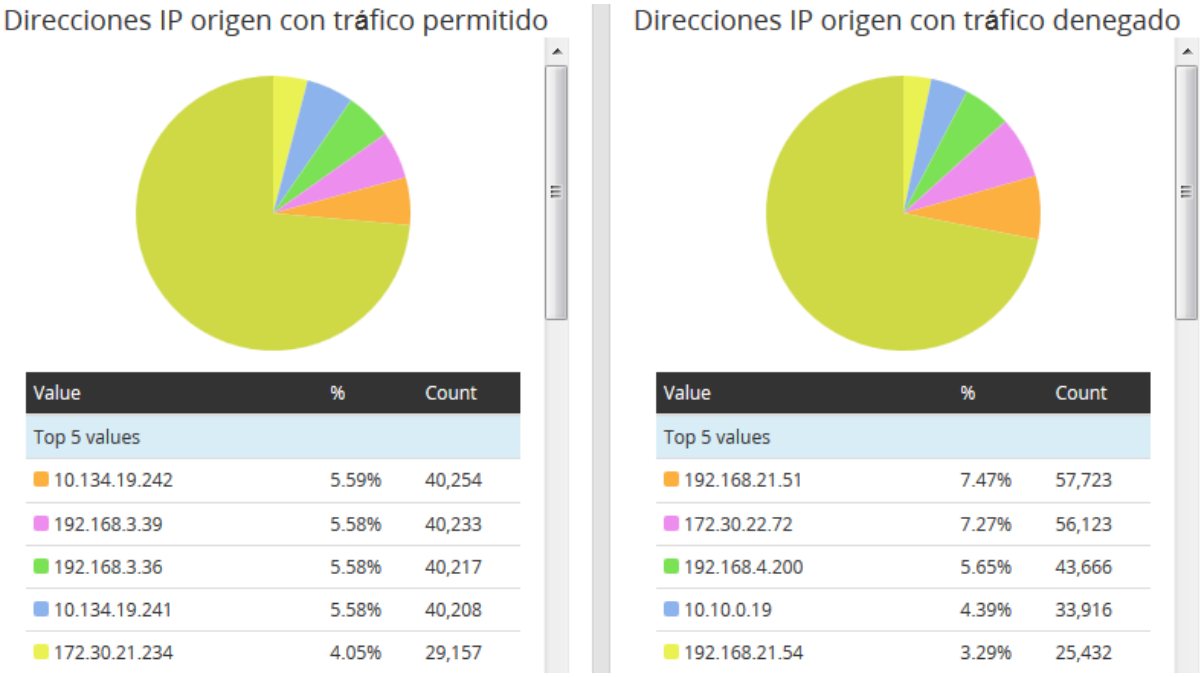


Figura 48: Direcciones IP origen con tráfico permitido y denegado

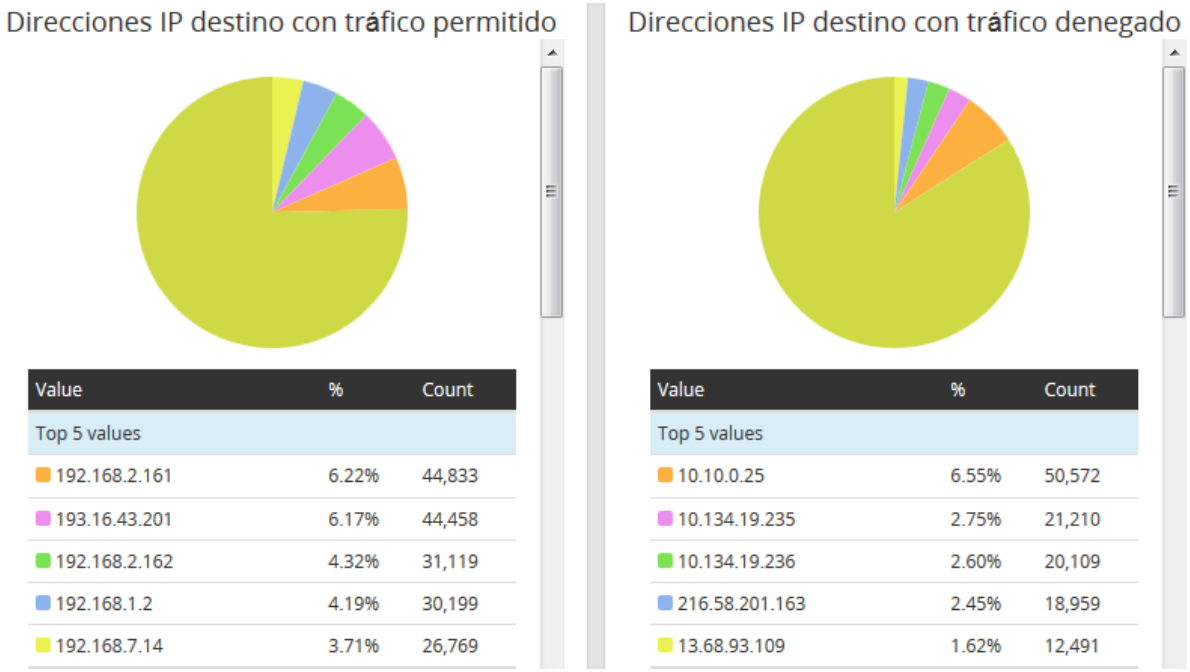


Figura 49: Direcciones IP destino con tráfico permitido y denegado

Toda esta información la hemos añadido al cuadro de mando para tenerla disponible en todo momento. En este trabajo los datos de cada gráfico no va a variar ya que estos datos se corresponden con los de los eventos enviados, pero carecemos de una fuente que los suministre de forma constante. Pero en caso de que tuviéramos un firewall enviando sus eventos a graylog en tiempo real si que variarían la

información que nos muestren conforme varíen los eventos recibidos al igual que ocurría en el caso práctico anterior.

Una vez tenemos esta información en el cuadro de mando podemos aprovechar la misma para realizar investigaciones sobre por qué está ocurriendo algo de lo que nos muestra. Por ejemplo, vemos que el principal motivo por el que finalizan las sesiones es porque se deniega el tráfico debido a alguna regla del firewall. Así que vamos a investigar esto.

Para ello, entramos en el stream Firewall y realizamos la siguiente búsqueda: sessionEnd:policy\-deny (el símbolo" \" se introduce para escapar el carácter "-",ya que este tiene el significado de exclusión en el lenguaje utilizado en las búsquedas). Una vez hemos realizado esta búsqueda vamos a ver que políticas son las que están denegando el tráfico:mediante RuleName:

Value	%	Count
Top 5 values		
interzone-default	99.93%	772,388
Rule-Deny-Reset	0.07%	521

Figura 50: Políticas que están denegando el tráfico

A continuación, vemos cuáles son las direcciones IP origen de dicho tráfico y observamos que son las mismas que tenemos en el cuadro de mando, por lo que la dirección IP origen con más tráfico denegado es 192.168.21.51. Así que vamos a ver que aplicación está utilizando, añadiendo esta IP a la búsqueda, quedando así: "sessionEnd:policy\-deny AND srcip:192.168.21.51".

Una vez realizada la búsqueda vamos a ver los puertos destino:

Value	%	Count
Top 5 values		
443	72.47%	41,831
80	27.37%	15,800
135	0.09%	50
445	0.06%	36
0	0.01%	6

Figura 51: Puertos destino de la IP examinada

De esto podemos concluir que principalmente se está denegando tráfico a los puertos 80 y 443 (que se corresponden con HTTP y HTTPS) por la aplicación de la regla interzone-default. Si esta situación no

fuera deseada habría que examinar las reglas del firewall para que deje de denegarse el tráfico. Además vemos que en 6 ocasiones se ha intentado establecer una conexión con el puerto 0, así que vamos a ver cuál era la dirección IP destino mediante la siguiente búsqueda:

sessionEnd:policy\-deny AND srcip:192.168.21.51 AND dstport:0

Si consultamos el campo dstip vemos que en los 6 casos la dirección IP destino es 192.168.23.7 y si vemos el campo Application vemos que es un ping. Así que si este comportamiento no fuera normal y tuviéramos acceso a ambos equipos podríamos tratar de averiguar a qué se deben dichos ping.

Además de este ejemplo podemos hacer cualquier tipo de consulta que se nos ocurra, sea a través de datos observados en el cuadro de mando o no. Por ejemplo, podemos ver si hay sesiones telnet (con puerto destino 23). Para ello hacemos la búsqueda dstport:23. Una vez hecha y viendo que hay resultados comprobamos las direcciones IP origen mediante el campo “srcip”:

Value	%	Count
Top 5 values		
10.134.19.5	51.26%	285
192.168.23.8	48.74%	271

Figura 52: Direcciones IP origen de tráfico telnet

Y mediante el campo “proto” comprobamos que en todos los casos es tráfico tcp. Así que a continuación comprobamos si el tráfico se permite o no mediante “action” y vemos que todos el tráfico telnet se ha denegado. Y mediante “Rule-Name” vemos que se ha denegado por la regla “interzone-default”.

4. Otras soluciones

Ya hemos visto a lo largo de este trabajo las posibilidades que nos ofrece graylog para el análisis de eventos de seguridad. Pero graylog no es la única opción existente en el mercado. En este capítulo examinaremos otras dos soluciones, Splunk y Elastic, mostrando cuáles son sus posibilidades y haciendo una comparativa con graylog. Además, examinaremos también las características que nos ofrece la versión de pago de graylog, graylog Enterprise.

4.1. Graylog Enterprise

Lo primero que tenemos que decir, es que graylog Enterprise es gratuito siempre y cuando el volumen de logs recibidos esté por debajo de los 5GB/día. Para obtener esta licencia gratuita hay que instalar la versión Open Source, obtener el Cluster ID de nuestra instalación y contactar con el departamento de ventas, dónde dando nuestros datos personales, de la empresa para la que trabajamos y este Cluster ID, nos proporcionarán una clave de licencia para disfrutar de la versión Enterprise de forma gratuita.

Graylog Enterprise dispone de todas las funcionalidades de la versión Open Source y además añade las siguientes:

- **Motor de correlación de eventos**: Permite analizar el conjunto de los eventos recibidos para determinar, en función de unas reglas que debemos establecer, si está ocurriendo un evento determinado. por ejemplo, si recibimos 100 intentos de login fallidos desde una mismo dirección IP y el siguiente intento de login tiene éxito este motor nos informaría que hemos recibido un ataque de fuerza bruta que ha tenido éxito.
- **Vistas y flujo de trabajo:** Esta funcionalidad es similar a la de Dashborads: nos permite ver información de manera gráfica. La diferencia radica en que mientras que en dashboards podemos ver paneles con información de una serie de campos obtenidos mediante una búsqueda que no varía, en las vistas y flujos de trabajo sí puede variar dicha búsqueda. Es decir, podemos crear una serie de paneles dinámicos que muestran valores y estadísticas que son el resultado de una búsqueda que incluirá una serie de campos con algún valor, pero podemos elegir en todo momento el valor de estos campos de la búsqueda sin necesidad de modificar los paneles. De esta forma, por ejemplo podemos crear paneles con información sobre una IP (que consultas DNS ha hecho, con que direcciones IP ha establecido una sesión, etc) y al entrar en ellos poder establecer la IP para la que queremos ver esta información.
- **Generación de informes**
- **Archivo offline de logs:** Esta funcionalidad nos permite guardar los mensajes recibidos en un fichero antes de eliminarlos, De esta forma los podemos mantener para el futuro y además tenemos la posibilidad de importarlos dentro de graylog en caso de ser necesario.

- **Soporte técnico**

4.2. Splunk

La primera alternativa a graylog para el análisis de eventos de seguridad que examinaremos es Splunk. En la actualidad Splunk se distribuye bajo cuatro licencias distintas: Splunk free, Light, Enterprise y Cloud. De estas cuatro opciones tan sólo es gratuita la primera, aunque también es la más limitada. Al igual que ocurría en graylog, todas las distribuciones de Splunk ofrecen un interfaz web a través del cual haremos uso de la aplicación. Las características de cada una de las distribuciones de Splunk son las siguientes:

Splunk Free es la única opción de la que podemos disponer de forma totalmente gratuita, aunque también es la más limitada en opciones, careciendo por ejemplo de la generación de informes. Además cuenta con dos limitaciones muy importantes: sólo permite tener un usuario y puede recibir como máximo 500MB de logs al día. Dispone de las siguientes características básicas:

- Recibir e indexar logs
- Almacenar métricas (timestamp, nombre, tamaño del evento recibido, etc).
- Realizar búsquedas
- Creación de cuadros de mando (dashboards) e informes.
- Parseo de datos
- creación de tablas y gráficos para visualizar los eventos.

La limitación en la cantidad de logs que puede recibir al día es la que más deberemos tener en cuenta en caso de valorar esta solución, ya que 500MB es una cantidad que se puede sobrepasar con facilidad.

Splunk light es una solución de pago orientada a pequeñas empresas. Esta solución es cómo la versión Enterprise excepto por las siguientes limitaciones:

- No dispone de Apps. Las apps son paquetes que se pueden añadir a Splunk, los cuales contienen un conjunto de configuraciones. De esta forma, añadiendo Apps podemos añadir funcionalidades que de otra forma tendríamos que crear nosotros.
- No es escalable, ya que tan solo permite el uso de un único servidor.
- No tiene control de acceso personalizable de forma que podamos dar distintos permisos a distintos usuarios, tan solo dispone de cuentas de administrador y de usuario
- Al igual que ocurría en Splunk Free también tiene limitado el número de usuarios y la cantidad de logs que puede recibir cada día. En este caso se pueden tener 5 usuarios cómo máximo y recibir hasta 20GB de logs diarios.

Como vemos, en caso de no requerir las funcionalidades de la versión Entrerprise de las que carece puede ser una opción a tener en cuenta. Eso sí, tenemos que tener cuidado con la limitación de 20GB de logs diarios, ya que en caso de sobrepasar dicha cantidad recibiremos un warning. Y en caso de acumular 5 warning en un periodo de 30 días no podremos realizar búsquedas hasta que hayan pasado 30 días desde que recibimos el primer warning. Por ello si se excede dicha cantidad no quedaría más remedio que utilizar la versión Enterprise.

En cuanto a Splunk Enterprise y Cloud ambas tienen las mismas funcionalidades, tan sólo cambia la forma en que utilizamos Splunk y cómo se cobra por el mismo.

Splunk Enterprise es una solución que deberemos instalar y mantener nosotros, por la que se nos cobra una licencia por el tiempo que utilicemos la solución.

Splunk Cloud es una solución SaaS, en la que se nos da acceso a una versión cluod de Splunk que utilizaremos como clientes, sin tener que preocuparnos del mantenimiento de la misma. En esta versión se nos cobrará según el volumen de datos que especifiquemos que se enviará diariamente, así como por servicios extra como el cifrado de los datos.

Las versiones light (con las excepciones mencionadas), Enterprise y cloud disponen, además de las funcionalidades de la versión free de las siguientes características:

- Permite configuraciones distribuidas, lo que permite realizar búsquedas de forma distribuida entre distintos nodos.
- Alta disponibilidad
- Recuperación en caso de desastre
- Posibilidad de crear alertas
- Control de acceso personalizable. Además, permite la utilización de sistemas LDAP y Single Sign-On para la autenticación.
- Soporte técnico
- Acceso completo a APIs y SDKs
- Posibilidad de añadir Apps.

Hay que decir que, al igual que Graylog, Splunk Enterprise dispone de una AMI en aws para poder probarlo de forma rápida (dispone de un periodo de prueba) o simplificar su despliegue si queremos utilizarlo en un entorno cloud al no tener que instalar nada. Pero no hay que confundir esto con la versión cluod. En caso de utilizar Splunk en aws simplemente estamos utilizando software en un

entorno cloud, pero el mantenimiento corre por nuestra cuenta (actualizaciones, copias de seguridad, etc). Por el contrario, al utilizar la versión cloud no nos tendremos que preocupar del mantenimiento.

4.2.1. Splunk vs Graylog

Una vez hemos visto las características de Splunk podemos compararlo con graylog. Como podemos ver, si optamos por una solución gratuita parece más interesante optar por la versión Open Source de graylog, ya que dispone de todas las características de Splunk free a excepción de la creación de informes. Pero por otro lado, en graylog podemos crear alertas y no vamos a tener limitación ni en el número de usuarios ni en el tamaño de logs que puede recibir diariamente

Si optamos por una solución Splunk de pago igualmente graylog puede ser una opción más interesante aún en el caso de optar por su versión gratuita, ya que gran parte de las características de las versiones de pago de Splunk las tenemos en la versión gratuita de graylog. Además, graylog permite clasificar los mensajes recibidos de forma más clara a través de los streams y pipeline, por lo que siempre vamos a poder elegir dónde enviar los mensajes que se reciban.

Y si comparamos la versión de pago de graylog con las de Splunk parecen soluciones muy similares, así que habría que pedir un presupuesto para ver cuánto nos costaría cada una de ellas puesto que sería un factor determinante para decantarnos por una u otra. Pero a priori graylog parece una solución más completa, además de poder utilizarla de forma gratuita si el volumen de los logs recibidos está por debajo de 5GB al día.

4.3. Elastic

Elastic es una empresa dedicada a desarrollar software de búsqueda, cuyo producto más destacado es Elastic Stack (anteriormente conocido como ELK Stack). Elastic Stack se compone de cuatro herramientas de las cuales la principal es Elasticsearch (de la que hace uso Graylog). Las otras tres herramientas que lo componen son Logstash, Kibana y Beats. La funcionalidad de cada uno de estos componentes es la siguiente:

- **Logstash:** Es un pipeline de procesamiento de datos. Es decir, este componente va a ser capaz de recibir datos de distintas fuentes, procesarlos, por ejemplo, perseando los datos, elimando o añadiendo información y después entrega estos datos ya procesados a su destino.
- **Beats:** Es una familia de agentes de recolección de datos. Estos agentes sirven para recoger los datos que hay en una serie de ficheros y enviarlos a su destino. Durante los casos prácticos utilizamos uno de estos agentes (Filebeat) para recoger los eventos y enviarlos a graylog.
- **Elasticksearch:** Es una base de datos no relacional en la que se almacenan los distintos eventos. Será sobre esta base de datos sobre la que realicemos las consultas.

- **Kibana:** Es el interfaz gráfico a través del cual podemos ver los datos que tenemos en ElasticSearch, permitiendo además visualizar los datos de forma gráfica a través de distintos paneles.

La arquitectura de Elastic Stack se puede ver en la siguiente imagen:

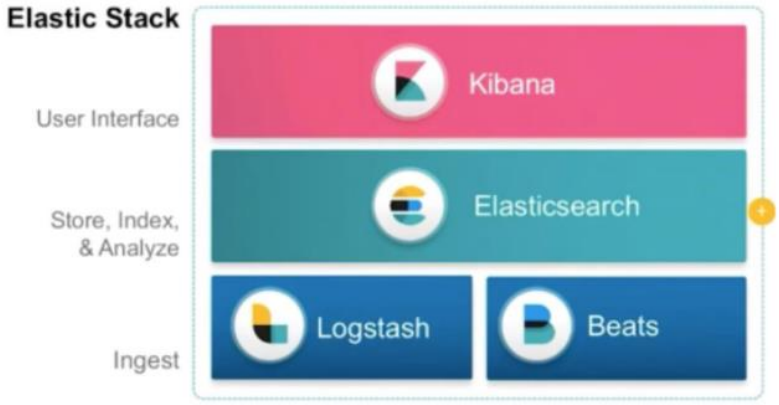


Figura 53: Arquitectura de Elastic Stack

Además de Elastic Stack, Elastic también dispone de otros productos orientados a la realización de búsquedas. Estos productos además se pueden utilizar junto a Elastic Stack ampliando así sus funcionalidades.

Elastic Stack, así como muchos de sus otros productos se distribuyen como software Open Source gratuito. Pero además de ofrecer el software de forma gratuita Elastic cuenta con un modelo de negocio basado en suscripciones. Al suscribirnos podemos optar por el plan básico, oro, platino o Enterprise. De estos planes el básico es gratuito, y el que estaremos suscritos de forma automática al instalar sus productos. Las otras tres suscripciones son de pago y cada una tiene todo lo incluido en la suscripción anterior más algunas funciones añadidas.
Además también dispone del servicio Elastic Cloud, a través del cual nos ofrecen sus productos como SaaS. Este servicio es de pago aunque ofrece una prueba gratuita de 14 días. Los servicios ofrecidos en Elastic Cloud son los mismos que ofrece como software tradicional que instalamos en nuestros sistemas.

Si optamos por hacer uso de Elastic Stack de forma gratuita, las principales características de las que dispondremos son las siguientes:

- Almacenamiento de datos en Elasticsearch sobre los que podremos realizar búsquedas.

- Realización de Snapshots de los datos almacenados en ElastiSearch
- Data rollups: Permite almacenar un histórico de los datos almacenados de forma similar a si se archivaran, pero de forma comprimida. Al hacer esto los datos ocuparán un espacio mínimo y aún podremos realizar búsquedas sobre ellos.
- Transformación de datos: permite crear un nuevo índice (los índices son las tablas de elasticsearch)) en Elasticsearch a partir de otro existente haciendo un tratamiento de los datos. Por ejemplo, si tenemos en un índice los eventos de un servidor DNS mediante una transformada podríamos tener otro índice que contenga el número de consultas que ha realizado cada dirección IP.
- Posibilidad de hacer despliegues distribuidos con balanceamiento de carga y alta disponibilidad.
- Comunicaciones encriptadas
- Control de acceso basado en roles
- Gestión de claves de API.
- Monitorización de Elastic Stack: Se puede monitorizar el estado de todos los componentes de Elastic Stack.
- API REST.
- Consola para interactuar con elasticsearch.
- Realización de búsquedas. Al escribir los términos de la búsqueda además se pueden hacer correcciones sobre los mismos, autocompletar, aplicar filtros y hacer sugerencias.
- Posibilidad de guardar búsquedas.
- Agregaciones de datos.
- Posibilidad de parsear los datos.
- Uso de plugins externos.
- Visualización gráfica de datos a través de paneles, pudiendo además compartirlos a través de un link o insertarlos en webs como un iframe.
- Visualizaciones geográficas de los datos, a través de **Elastic Maps**
- Uso de Canvas, con el que podemos elegir como visualizar y presentar los datos, eligiendo por ejemplo, el color y tamaño de los textos, la fuente que utilizarán o el fondo que tendrá la pantalla de visualización.

Con la suscripción oro, se disponen además de las características de la suscripción básica, de las siguientes características:

- En Elastic ELK se dispone de filtrado de IP y autentificación mediante LDAP o Active Directory.
- Monitorización centralizada de múltiples nodos de Elastic Stack desplegados de forma distribuida.

- Configuración del periodo durante el que se almacenarán los datos de monitorización (por defecto 7 días).
- Configuración de alertas de monitorización de los componentes de Elastic Stack.
- Soporte técnico.

Las características que se añaden en caso de optar por la suscripción platino son las siguientes:

- Replicación entre cluster. Esta característica permite replicar los índices de un cluster en otro. De esta forma obtenemos recuperación en caso de desastre.
- Opciones avanzadas de seguridad. Concretamente se dispondrá de control de acceso basado en atributos, single sign-on, seguridad a nivel de campo y documento y la posibilidad de utilizar realms de autenticación y autorización personalizados.
- Machine Learning
- Soporte mejorado.

Finalmente, con la suscripción Enterprise obtendremos además de todo lo anterior las siguientes características:

- Uso de Elastic Endpoint Security. Este es un producto de seguridad para puntos finales similar a un antivirus con la ventaja de que se puede integrar totalmente en Elastic SIEM. De esta forma se puede centralizar la administración de la protección de todos los puntos finales desde Elastic SIEM así como poder obtener información detallada de lo que ocurre en cada uno de ellos.

Además, tanto si optamos por la suscripción gratuita como por alguna de pago, podemos hacer uso de las siguientes herramientas además de Elastic Stack:

- **Elastic APM**. Esta es una herramienta de análisis de métricas con la que es posible analizar métricas de aplicaciones.
- **Elastic Uptime**. Con esta herramienta se puede monitorizar los tiempos de actividad de servicios y dispositivos. De esta forma es posible saber cuándo están activos y observar si tienen algún problema que haga que no respondan correctamente.
- **Elastic SIEM**. Esta herramienta la podemos usar para hacer una gestión de información y eventos de seguridad, analizando por ejemplo los eventos de autenticación, el tráfico DNS o viendo el netflow, es decir, desde dónde y hacia dónde se está generando el tráfico de red.
- **Elastic App search**. Esta es una herramienta de búsqueda basada en Elasticsearch que permite realizar búsquedas dentro de aplicaciones y visualizar los datos de forma gráfica. Permite realizar búsquedas sobre cualquier tipo de dato y configurar el motor de búsqueda, por

ejemplo, creando listas de sinónimos, asignando ponderaciones o reorganizando los resultados.

4.3.1. Elastic vs Graylog

Como hemos visto Elastic ofrece un buen número de herramientas que podemos utilizar para el análisis de eventos de seguridad, siendo su herramienta principal Elastic ELK. Tal y como hemos visto tanto Elastic ELK como la práctica totalidad de sus herramientas se pueden utilizar de forma gratuita. Si nos ceñimos exclusivamente al uso de las herramientas de forma gratuita podemos ver que el conjunto de sus herramientas nos otorga unas funcionalidades más completas que graylog. Además cuenta con la ventaja de que podemos instalar sólo las herramientas que necesitemos.

En caso de que optáramos por una suscripción de pago obtendríamos un conjunto de funcionalidades más completas que Graylog Enterprise. Pero además contamos con la ventaja de tener tres tipos de suscripciones distintas, de forma que podemos elegir aquella que más se ajuste a nuestras necesidades, tanto en función de las funcionalidades que nos ofrece como del precio que tenga cada una de estas suscripciones. Por el contrario graylog Enterprise no se ofrece de forma gradual para poder elegir un tipo de licencia que aunque no nos de acceso a todas las funcionalidades, si nos de las que necesitemos y cueste menos dinero.

5. Conclusiones y líneas futuras de trabajo

Durante este trabajo hemos podido ver que el uso de una herramienta para el análisis de eventos de seguridad es imprescindible si queremos saber lo que ocurre en nuestros sistemas. De no haber usado ninguna herramienta para analizar los eventos de los casos prácticos nos hubiera sido imposible realizarlos. Esto ha sido particularmente notorio en el caso de los eventos del firewall, dónde hemos analizado una muestra de más de 1600000 eventos y buscar algo en ellos con un simple editor de texto es algo prácticamente imposible. Sin embargo al utilizar una herramienta que nos permite parsear los eventos como deseemos, y utilizar los campos en los que almacenamos su información para realizar búsquedas, no sólo nos ha permitido que esta sea una tarea sencilla, si no además rápida. Además, el uso de una herramienta no nos permite sólo realizar las búsquedas, si no presentar resultados de forma gráfica y crear paneles a través de los que podemos obtener información en todo momento con un solo vistazo.

En cuanto a graylog, que ha sido la herramienta con la que hemos realizado el análisis de los eventos, hemos podido ver en primer lugar que resulta muy sencillo enviar los eventos de seguridad y tratarlos para obtener información de ellos. Una vez hemos tratado los eventos también hemos visto la facilidad con la que henos podido resolver casos prácticos de uso. Además hemos visto cómo es posible centralizar los eventos de distintas fuentes (en los casos prácticos de uso SSH, DNS, WAF y firewall), pudiendo enrutar los mensajes de cada una de las fuentes a un stream distinto para tener así los eventos correctamente clasificados.

Finalmente, hemos visto que graylog no es la única solución y hemos hecho un análisis de Splunk y Elastic. Splunk no es una solución factible en caso de querer utilizar una solución gratuita. Pero Elastic sí que ofrece una buena solución Open Source gratuita (Elastic Stack) la cual puede ser una buena alternativa a graylog. Además esta solución se puede combinar con otras soluciones de búsqueda de Elastic para cubrir todas nuestras necesidades.

En cuanto a líneas futuras de trabajo que pueden realizarse de forma adicional a este trabajo podríamos hacer lo siguiente:

- Obtener una licencia gratuita de graylog Enterprise para evaluar las características que añade esta versión.
- Desplegar Elastic Stack con el objetivo de desarrollar en él los casos prácticos de uso expuestos en este trabajo. Una vez realizados se podría hacer una comparativa con graylog desde la experiencia obtenida con ambas soluciones (facilidad de despliegue de la solución, facilidad de uso, características, etc). De esta forma la comparativa sería mucho más completa

y exacta que la expuesta en este trabajo, dónde únicamente hemos considerado las características que tiene Elastic de forma teórica con la información recopilada. Pero no hemos utilizado dicha solución para comprobar si su funcionamiento es tan sencillo e intuitivo como graylog ni sabemos si los resultados serían satisfactorios desde el punto de vista de la usabilidad, tiempo de respuesta a consultas y presentación de resultados.

Bibliografía

Documentación sobre Amazon EC2 (consultado a 29/10/2019):
https://docs.aws.amazon.com/es_es/AWSEC2/latest/UserGuide/concepts.html

Documentación sobre Graylog (consultado a 29/10/2019):
http://docs.graylog.org/en/3.1/index.html

Detección de Botnets basada en DNS (INCIBE-CERT) (consultado a 25/11/2019):
https://www.incibe-cert.es/blog/botnets-dns

WAF: cortafuegos que evitan incendios en tu web (consultado a 25/12/2019):
https://www.incibe.es/protege-tu-empresa/blog/waf-cortafuegos-evitan-incendios-tu-web

Web de Splunk (consultado a 4/12/2019): https://www.splunk.com/

Características de las distintas versiones de Splunk (consultado a 4/12/2019):
https://www.splunk.com/en_us/software/features-comparison-chart.html

Características de Splunk Light (consultado a 4/12/2019):
https://docs.splunk.com/Documentation/SplunkLight/7.3.2/GettingStarted/AboutSplunkLight

Características de Splunk Enterprise (consultado a 4/12/2019):
https://docs.splunk.com/Documentation/Splunk/6.4.1/Overview/AboutSplunkEnterprise

Guía ràìda de Splunk (consultado a 4/12/2019):
https://www.splunk.com/pdfs/solution-guides/splunk-quick-reference-guide.pdf

Web de Elastic (consultado a 8/12/2019): https://www.elastic.co/es/

Suscripciones de Elastic (consultado a 8/12/2019): https://www.elastic.co/es/subscriptions

Anexos

Anexo A: Configuración de Sidecar

Una vez instalado Sidecar y filebeat en cada una de las instancias de Ubuntu he seguido los siguientes pasos:

1- Obtener un API token: Para ello he ido a System/Authentication management y he pinchado sobre “edit tokens” en mi usuario.
2- En la ventana que se muestra he creado un token llamado sidecar_token

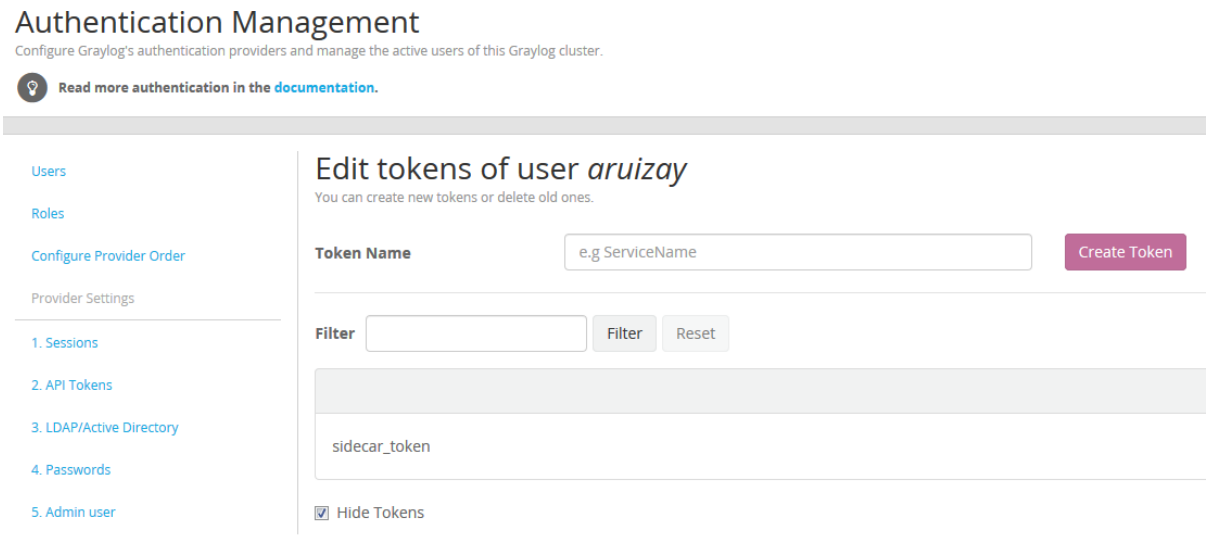


Figura 54: Token de graylog

3- En cada una de las instancias Ubuntu he editado el fichero */etc/graylog/sidecar/sidecar.yml* donde los principales valores que hay que introducir son los siguientes:
   - Server_url: Hay que poner http://direccionip_graylog:9000/api
   - Server_api_token: hay que introducir el API token creado previamente
   - Node_name: el nombre que queremos dar al nodo. Es como aparecerá en la consola web de Graylog en el apartado System/Sidecar
4- Arrancamos los Sidecar. Nos deben aparecer en la consola web de Graylog en System/Sidecar con el nombre que le hayamos dado al nodo.
5- En esta ventana vamos a Configuration y pinchamos en “Create Configuration”

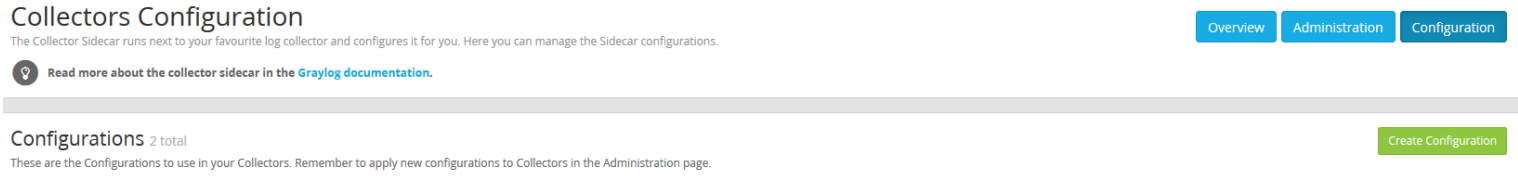


Figura 55: Configuración de colectores

6- Damos el nombre que deseemos a la configuración y elegimos el colector que usaremos, en este caso filebeat en Linux. Nos aparecerá un fichero de configuración de filebeat predeterminado que en muchos casos podremos usar. En este caso tan solo cambiamos el campo “hosts” para que la IP se corresponda con la de graylog y el campo field.source para que sea el nombre que hemos dado al nodo

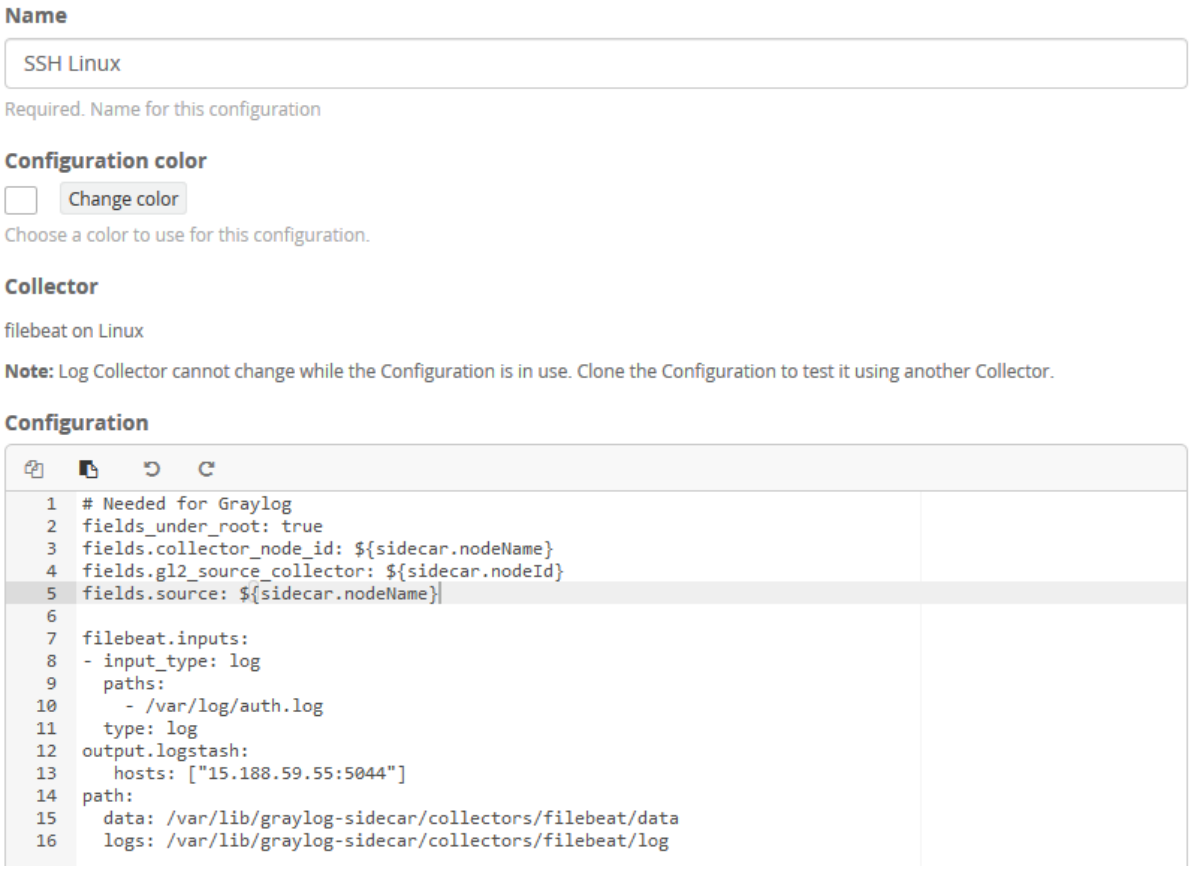


Figura 56: Configuración de filebeat

7- Vamos a “Administration”. Aquí nos aparecerán todos los sidecar que tengamos. Para cada uno, debemos elegir el tipo de colector que usaremos (por defecto sólo nos aparecen filebeat o nxlog, pero podemos crear otros). Seleccionamos filebeat y pinchamos en Configure donde nos aparecerán las configuraciones disponibles para ese colector. Seleccionamos la que hemos creado antes y la asociamos al Sidecar. Veremos como aparece el nomber de la configuración en cada Sidecar al que se la hayamos asociado:

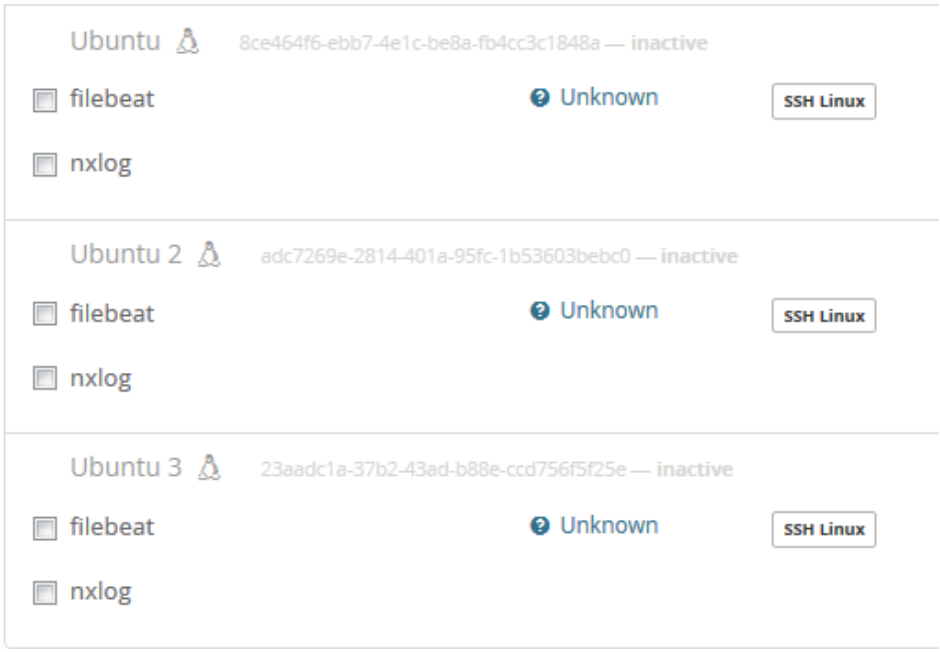


Figura 57: Colectores

8- Con esto ya deberíamos recibir los mensajes de cada instacia Ubuntu. Lo podemos comprobando desde System/Sidecars dónde podemos pinchar sobre “show messages” de cualquier sidecar para ver los mensajes que hemos recibido desde ese sidecar.