Proyecto Fin de Carrera
Ingeniería de Telecomunicación

Detección de anomalías en los registros de tráfico ofrecidos por IPFIX

Autor: Agustín Walabonso Lara Romero

Tutor: Rafael Estepa Alonso

Dpto. de Ingeniería Telemática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2018

Proyecto Fin de Carrera
Ingeniería de Telecomunicación

Detección de anomalías en los registros de tráfico ofrecidos por IPFIX

Autor:
Agustín Walabonso Lara Romero

Tutor:
Rafael Estepa Alonso
Profesor titular

Dpto. de Ingeniería Telemática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2013

Trabafo Fin de Grado: Detección de anomalías en los registros de tráfico ofrecidos por IPFIX

Autor: Agustín Walabonso Lara Romero

Tutor: Rafael Estepa Alonso

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2018

El Secretario del Tribunal

A mi familia

A mis profesores

A mis amigos

Agradecimientos

En primer lugar, quiero agradecer a mis padres, Agustín y Maria Victoria por haberme permitido realizar mis estudios, formarme como persona y apoyarme en todo lo que me he propuesto.

En segundo lugar, agradacer a todas las personas con las que paso mucho tiempo y siempre están conmigo, Manuel Ferrera, Andrés, Ismael, Juan, Curro, Juan Angel, Francisco, Lourdes. En definitiva, a todos mis compañeros de clase por haber estado tan unidos, siempre disponibles para ayudar al que lo necesitaba.

En tercer lugar, agradecer a toda mi familia, con especial cariño a mi abuela, por todo el gran apoyo que me han brindado.

Gracias también a Mónica Pérez por estar todos los días ayudándome en todo, dándome un gran apoyo y saber que siempre estás ahí para todo.

Gracias a todos mis profesores por toda la formación y conocimiento proporcionado a lo largo de mi etapa de estudiante.

No se me olvida agradecer a mis compañeros de Everis el haberme permitido trabajar con ellos y enseñarme tanto en mis prácticas de empresa ¡Espero que podamos volver a trabajar juntos!

También agradecer a mis grandes compañeros de Hapkido que somos una gran familia.

Por último y no menos importante, agradecer al departamento de telemática por haber conseguido motivarme y querer trabajar en el mundo de la telemática. En especial, agradecer a mi tutor de proyecto, Rafael Estepa, por ayudarme y permitirme llevar a cabo este proyecto, así como motivarme en el mundo de la ciberseguridad. Gracias por su asesoramiento y corrección.

Agustín Walabonso Lara Romero

Sevilla, 2018

Resumen

El de la ciberseguridad es un campo que siempre está evolucionando. Esto hace que sea uno de los escenarios más complejos, pues los enfoques y paradigmas cambian continuamente.

Cada año se encuentran un gran número de vulnerabilidades nuevas en los diferentes sistemas, y es por ello por lo que constantemente se desarrollan técnicas y software que eliminan, bloquean ó identifican estas vulnerabilidades.

El objetivo de este documento es la investigación y desarrollo de técnicas de detección de comportamientos anómalos en el tráfico de red, así como la realización de pruebas que permitan valorar el funcionamiento de los mismos.

Abstract

Cybersecurity is a constantly evolving field. That makes it particularly complex since its approaches and paradigms change frequently. Each year, a large number of new vulnerabilities across different systems is identified. To tackle these, new techniques and software are constantly being developed as to eliminate, block or identify these vulnerabilities. This document's objectives are to research on and develop techniques to identify anomalous behaviour in network traffic, as well as running performance tests allowing for the evaluation of their functioning.

Índice

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

1 INTRODUCCIÓN Y OBJETIVOS

> Creo que los virus informáticos deberían contar como vida. Creo que dice bastante sobre nosotros el hecho de que la única forma de vida que hemos logrado crear sea puramente destructiva. Hemos creado vida basada en nuestra imagen.
>
> *- Stephen Hawking-*

1.1 Introducción

En una red se realizan múltiples conexiones, son en estas dónde podemos detectar diferentes intrusiones ó software anómalos. Es por ello por lo que se realizan técnicas de tratamientos matriciales del tráfico. Dichas técnicas ofrecen la posibilidad de realizar estudios matemáticos con el fin de poder detectar anomalías en la información cursante de las redes.

Debido al gran número de vulnerabilidades existenten en las diferentes redes, justifica la gran necesidad del desarrollo continuo de sistemas capaces de llevar a cabo trabajos de detección de las mismas.

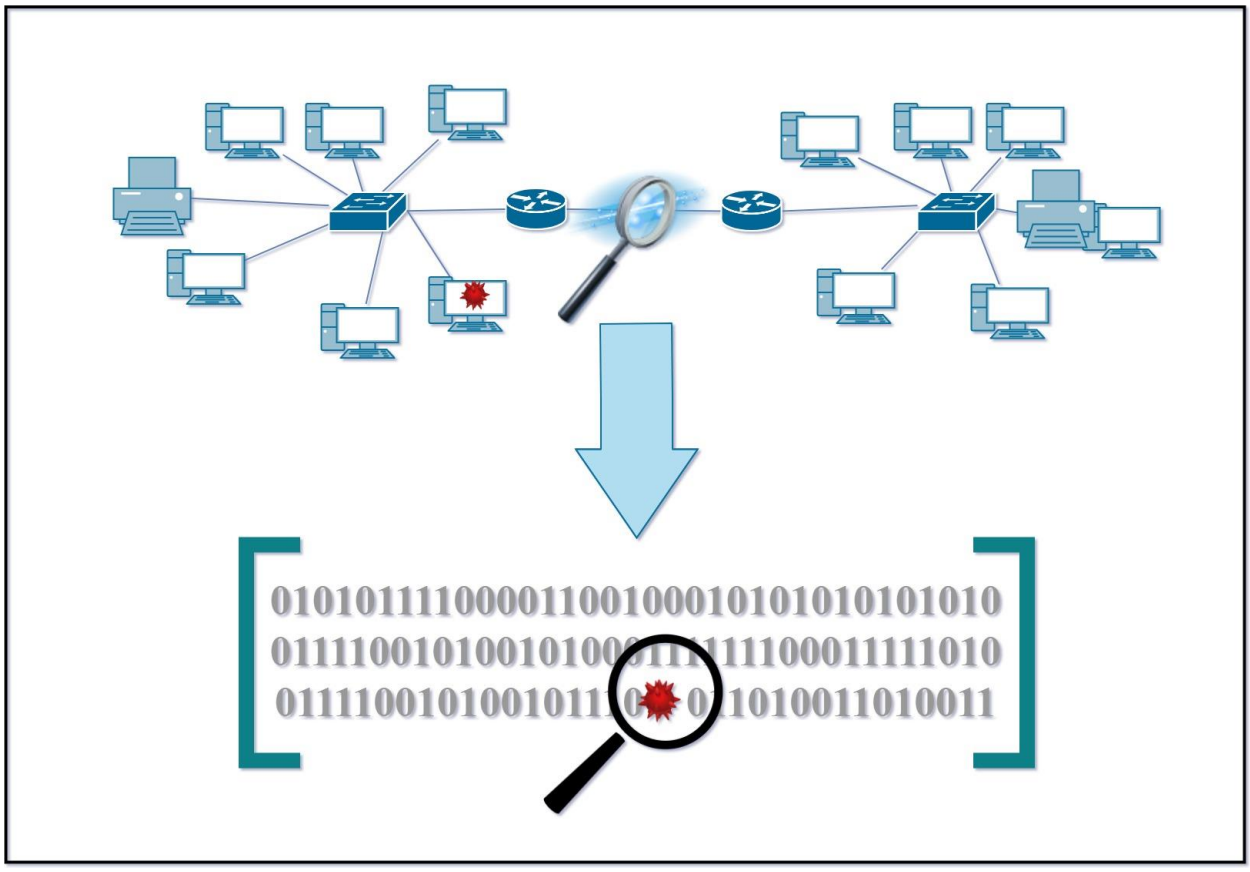

Figura 1-1 Supervisión de matrices de tráficos ofrecidos en las redes.

Un usuario que hace uso de dispositivos conectados a la red realiza una serie de acciones cotidianas que

pueden ser clasificadas y guardadas como patrones de comportamiento. Sobre estos, pueden llevarse a cabo estudios y toma de decisiones para comprobar la existencia de anomalías.

Las técnicas de detección de anomalías se basan en la clasificación y estudio de patrones. Cuando se identifica un patrón que contiene valores que difieren de su modelo, el sistema lo detectará como anómalo por lo que, a posteriori, se realizará un trabajo de supersión para comprobar si resulta ser un falso positivo o, por lo contrario, se trata realmente de la detección de una anomalía.

Cabe destacar que estos sistemas siempre suelen alertar de falsos positivos, por lo que es muy importante llevar a cabo un buen estudio previo sobre las condiciones a trabajar y así poder configurar e implantar un modelo acorde tanto a las necesidades como a la escabilidad y el entorno sobre el que se desea operar.

En una red de datos existe un intercambio continúo de flujos y en el presente trabajo se exponen modelos estadísticos así como la clasificación del flujo para la localización de anomalías.

1.2 Motivación

Para la detección de anomalías es necesaria la manipulación de una inmensa cantidad de información, de modo que para su procesamiento es necesario el uso de modelos estadísticos adecuados.

Resulta de gran interés realizar un estudio profundo de los paquetes que viajan por la red, pues esto permite obtener una información útil para ser estudiada y poder definir comportamientos habituales.

En la actualidad existen colectores netflow que recogen la información que viaja por la red. NetFlow es un protocolo de red diseñado para recolectar la información sobre tráfico IP, la cuál hace posible la monitorización de las redes.

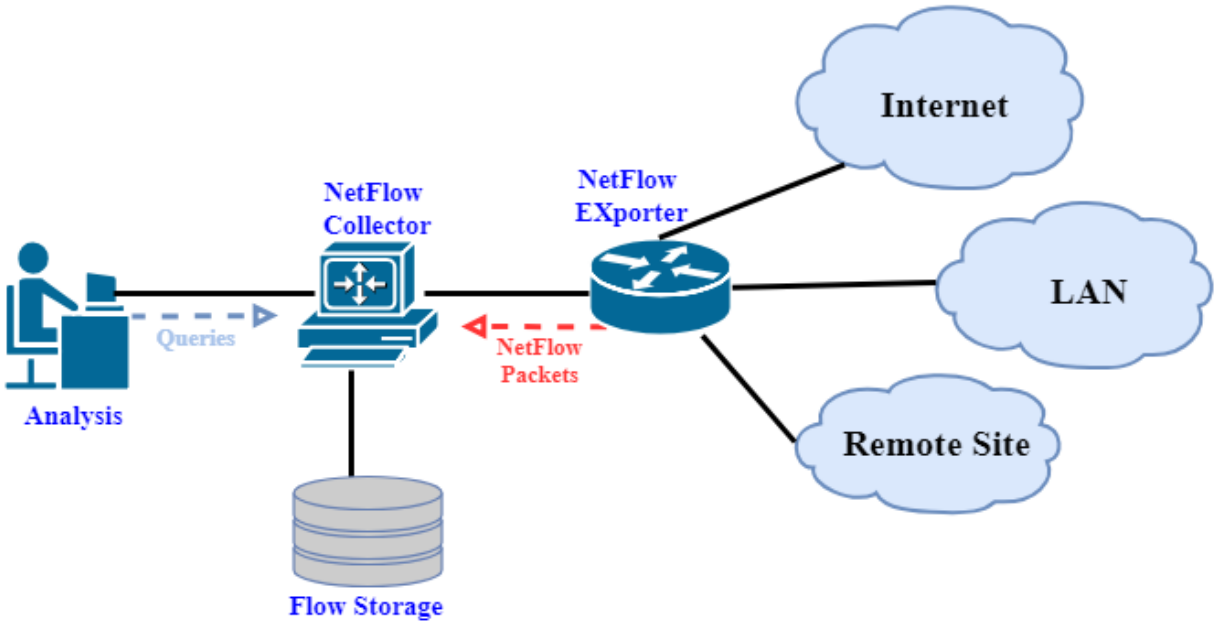

Figura 1-2 Funcionamiento de Netflow

Debido a las vulnerabilidades existentes en la red, mucho del tráfico cursado se encuentra cifrado. Por ejemplo, un usuario hace uso de una aplicación web que utiliza el protocolo HTTPS (Protocolo seguro de transferencia de hipertexto). Este es un protocolo de transferencia de datos de forma segura, es decir, el contenido que transporta se cifra, esto permite que si un atacante consiguiera hacerse con estos datos, dicho atacante, a priori, no podría interpretar la información substraída.

Figura 1-3 Ubicación en el modelo de capas OSI.

Debido a que el tráfico se transporta de forma cifrada se crea un gran interés de la realización de inspección profunda de paquetes, para poder identificar la aplicación que se encuentra detrás del tráfico cifrado.

1.3 Metodología de trabajo

Para el desarrollo del presente trabajo se ha llevado a cabo la metodología de trabajo Scrum. En la cuál se recogen un conjunto de buenas practicas para trabajar y llevar a cabo la realización de un Proyecto.

Esta metodología de trabajo se basa en la división del proyecto en una lista de tareas pequeñas ordenadas por prioridad y puntos de esfuerzos. Estas pequeñas tareas pasan por distintas etapas hasta la finalización de las mismas.

También se ha optado por el uso de GitHub como repositorio del proyecto. Esto hace posible que se pueda llevar un buen control de versiones y cambios en el software desarrollado.

Para poder unificar GitHub y la metodología de trabajo se ha decidido trabajar con waffle. Esta es una herramienta de uso libre que permite la planificación de un proyecto así como la sincronización del mismo en GitHub.

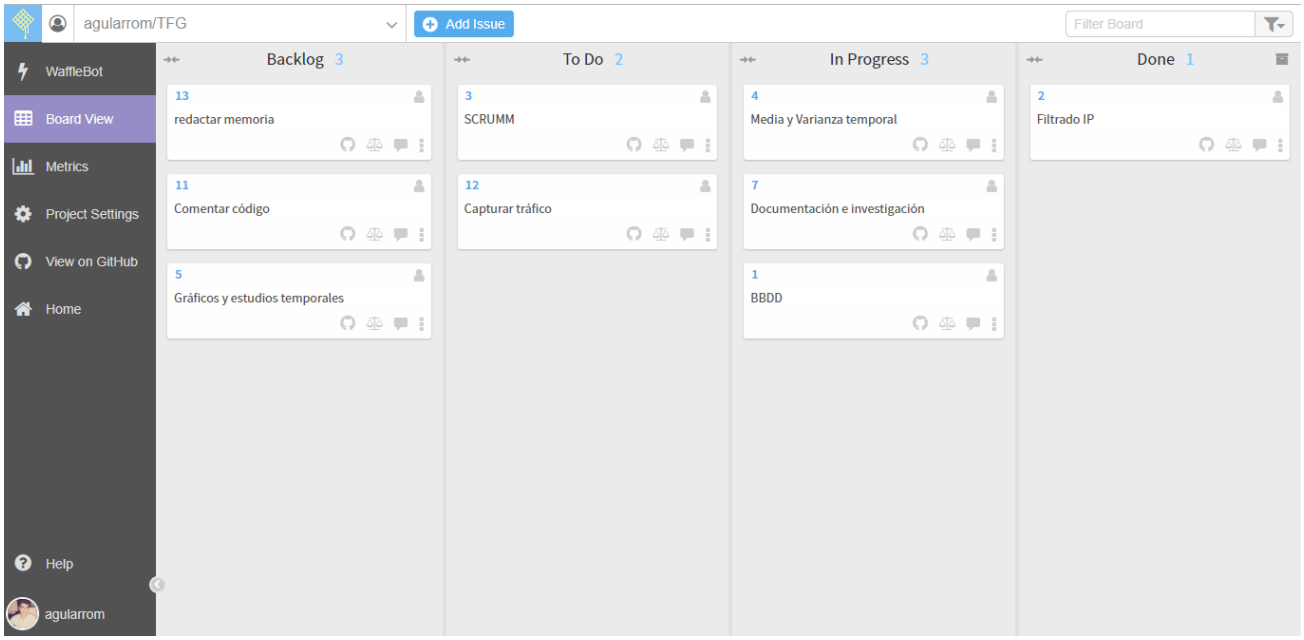

Figura 1-4 Panel de waffle.io

1.4 Objetivos

El enfoque de este trabajo consiste en el desarrollo de un sistema autónomo basado en la investigación estadística para solventar los problemas de detecciones de anomalías en redes de datos.

Los objetivos específicos de este trabajo se categorizan en el siguiente orden:

- Análisis de las diferentes técnologias similares existentes.
- Análisis de librerías de software libre que realicen inspección profunda de paquetes de red y permitan identificar aplicaciones en el flujo de red.
- Análisis de los diferentes recolectores de flujo de red.
- Análisis de modelos estadísticos para la clasificación y toma de decisiones.
- Pruebas y funcionamiento de librerías.
- Desarrollo de un sistema real y escalable que pueda ser usado como método de defensa, así como la posibilidad de poder ser implantado en cualquier lugar.
- Búsqueda y generación de dataset para ser usado como entrada al modelo.
- Planteamiento de un escenario real.
- Instalación y configuración del escenario.
- Pruebas experimentales.
- Análisis y validación de los datos obtenidos.
- Conclusiones y posibles líneas de mejora.

2 ESTADO DEL ARTE

Una vez un ordenador me venció jugando al ajedrez, pero no me opuso resistencia cuando pasamos al kick boxing

- Emo Philips -

Este capítulo recoge el estudio de diferentes detectores de anomalías en tráfico de red, así como una introducción de las vulnerabilidades existentes en las redes. Para finalizar, se recoge una conclusión sobre las diferentes tecnologías mencionadas y la dirección que llevan.

2.1 Vulnerabilidades en la red.

Es muy fácil confundir vulnerabilidad con amenaza. Una vulnerabilidad se define como el fallo en un sistema de información. Dicho fallo compromete la integridad del sistema. Mientras que por otro lado, una amenaza es aquella que aprovecha una vulnerabilidad para acceder a un sistema de información con fines maléficos.[1]

Según el instituto de ciberseguridad de España, el riesgo en un sistema de información se encuentra en la intersección de la existencia de una vulnerabilidad, una amenaza y el propio sistema.

Figura 2-1 Definición de riesgo en los sistemas de información.[1]

En la red hay multiples vulnerabilidades ya sea inalámbrica o cableada. A continuación se explican diferentes ataques posibles que pueden ser llevados a cabo a través de una persona o empresa con fines maléficos.

2.1.1 Man In The Middle.

Este ataque (hombre en el medio) es una técnica que permite situar al equipo atacante en medio de la comunicación del equipo de la víctima y el router. Esto hace que el atacante substraiga información de la víctima. Este tipo de ataque resulta muy eficaz cuando la información que se transmite no se encuentra cifrada, puesto que, en caso contrario el atacante tendría que realizar técnicas de desencriptado para que la información substraída pueda ser legible. [2]

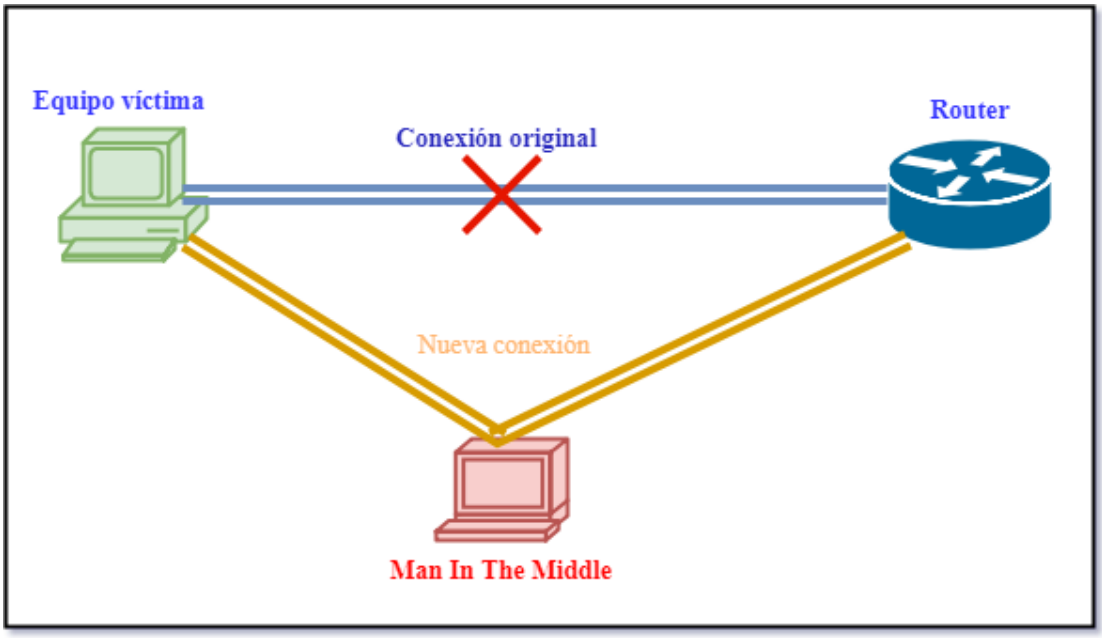

Figura 2-2 Ataque Man In The Middle

2.1.2 Denegación de servicios (DOS).

Este ataque es uno de los más conocido y extendido. Básicamente consiste en dejar inaccessible para los usuarios un servicio o recurso durante un cierto tiempo.

Este ataque suele ser usado como distracción para los administradores de la red y así poder realizar otro ataque más potente y con un claro objetivo.

Llama la atención que el 80% de la veces que se realiza este ataque resulta ser desde el interior de la red [3].

Un DOS típico, es la de dejar sin recursos a un servidor inundándolo a peticiones falsas con el fin de dejar inaccesible el servicio que ofrece.

Para conseguir que este ataque sea muy eficaz. Los atacantes suelen infectar una red de ordenadores para poder realizar un DOS masivo.

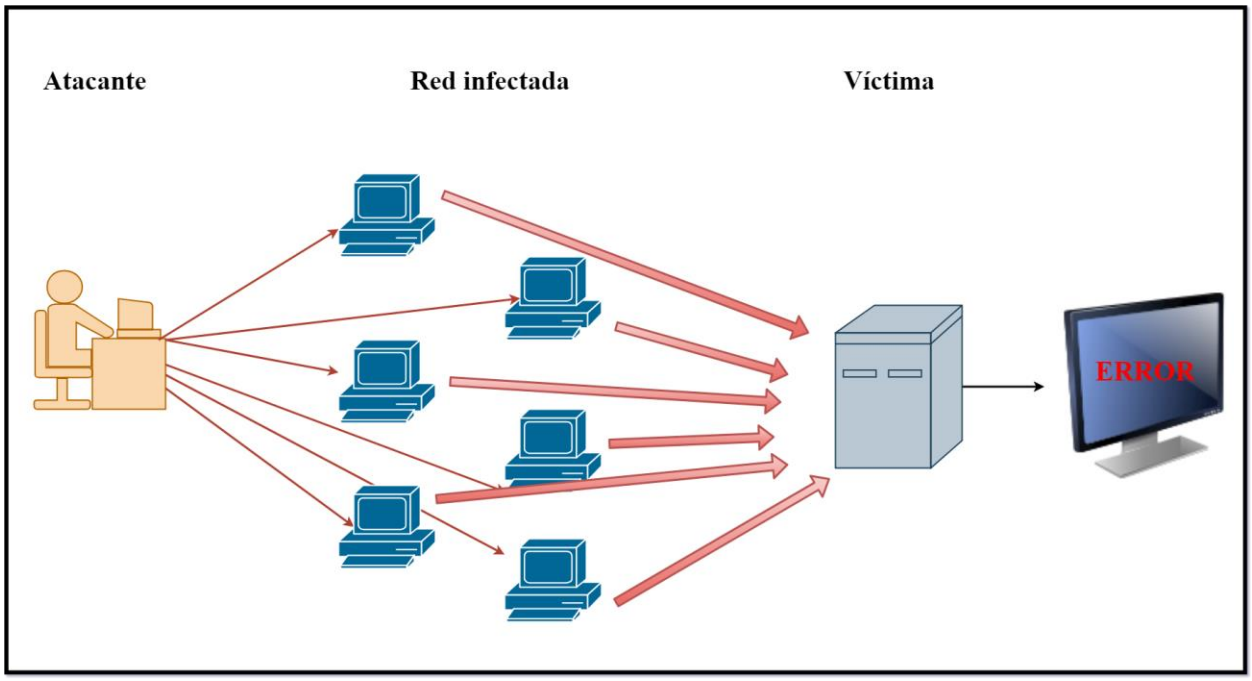

Figura 2-3 Ataque de denegación de Servicios

2.1.3 Ransomware.

Se clasifica este ataque como un malware capaz de acceder a documentos de la víctima con el fin de impedir el acceso a los mismos. El fin de este ataque es el de pedir un rescate por la recuperación de los documentos.

Una de las propiedades que tiene es la facilidad de propagación. Uno de los medios utilizados más habituales es el envío de correos maliciosos. [4]

El rescate siempre se exije en bitcoins, ya que es una criptomoneda que hace que sea difícil seguir el rastro de la trasacción llevada a cabo.

Para que el ataque sea efectivo los hackers realizan mucha ingeniería social para aprovechar las brechas de seguridad.

A continuación se citan algunas de las diferentes técnicas realizadas para conseguir infectar con el malware Ransonware:

- Obtención de cuentas con privilegios de administrador.
- Envío de spam.
- Suplantación de identidad para descargar un archivo infectado.

Figura 2-4 Ramsonware

2.2 Detección de anómalias en la red.

Actualmente existen numerosas aplicaciones, librerías y técnicas para lograr detectar anomalías en el tráfico circulante. Para llevar a cabo esta tarea es muy importante realizar modelos estadísticos de tráfico limpio. Debido a que un modelo un incorrecto implica errores en la detección. Pudiendo causar un alto número de falsos positivos ó incluso la no detección de positivos. [5]

2.2.1 Redes neuronales.

Los sistemas inteligentes cada vez sirven más de ayuda para dar solución a problemas cotidianos en todos los campos. Por ejemplo, en el mundo de la ciberseguridad son muy importantes para la detección de anomalías. Actualmente existen trabajos que hacen uso de redes neuronales para la predicción del tráfico [6], [7]. Esta labor se consigue a través del entrenamiento de un modelo neuronal del tráfico circulante en una red.

Estos sistemas contien son formados por neuronas divididas en tres capas [8]:

- Una primera capa de entrada que introduce los patrones al sistema.
- Una segunda capa denominada como ‘capas ocultas’ que son formadas por neuronas interconectadas entre sí formando multicapas.
- Por úlitmo una tercera capa constituída por los valores de salidas del sistema.

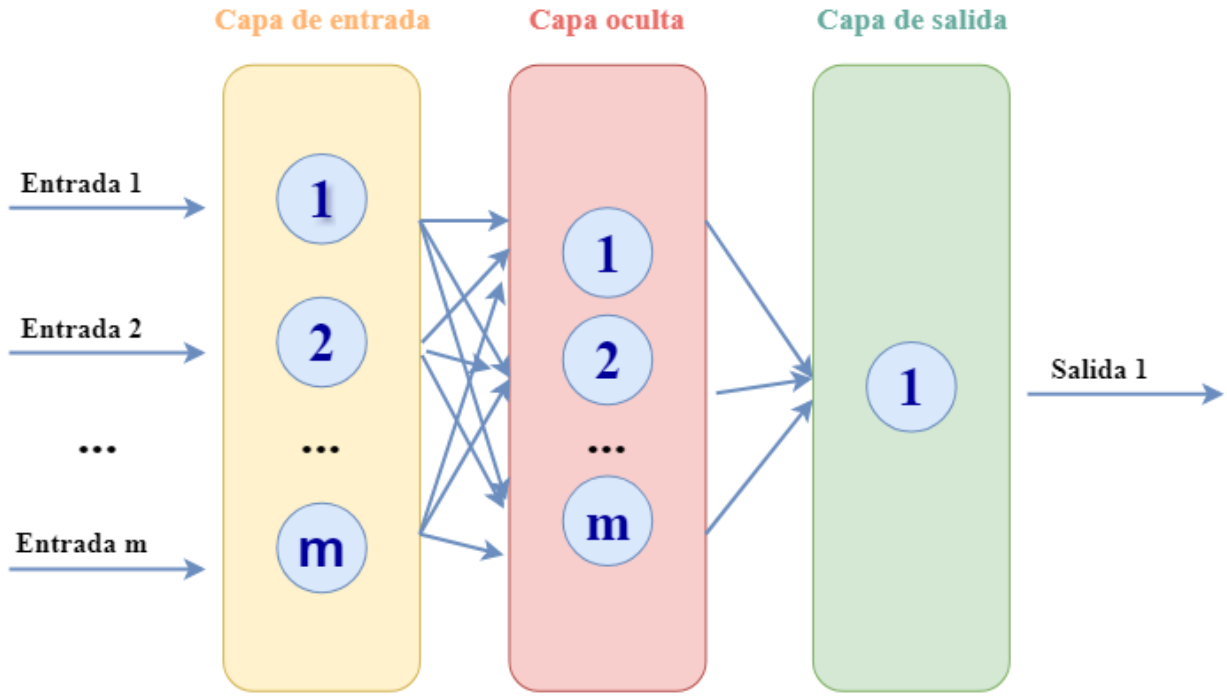

Figura 2-5 Modelo de red neuronal

2.2.2 R&S®PACE 2.

PACE es una librería software capaz de detectar múltiples aplicaciones y protocolos de red haciendo uso de diferentes técnicas de detección: [9]

- Inpección profunda de paquetes.
- Análisis de comportamiento heurístico y estadístico.

Esta librería es capaz de detectar aplicaciones y extraer metadatos de tráfico cifrado. Para conseguir todo esto, el software realiza los siguientes pasos:

1. Preparación de los paquetes.
2. Reordenación de los paquetes.
3. Clasificación de paquetes.
4. Decodificación.
5. Tiempo de espera de manejo.

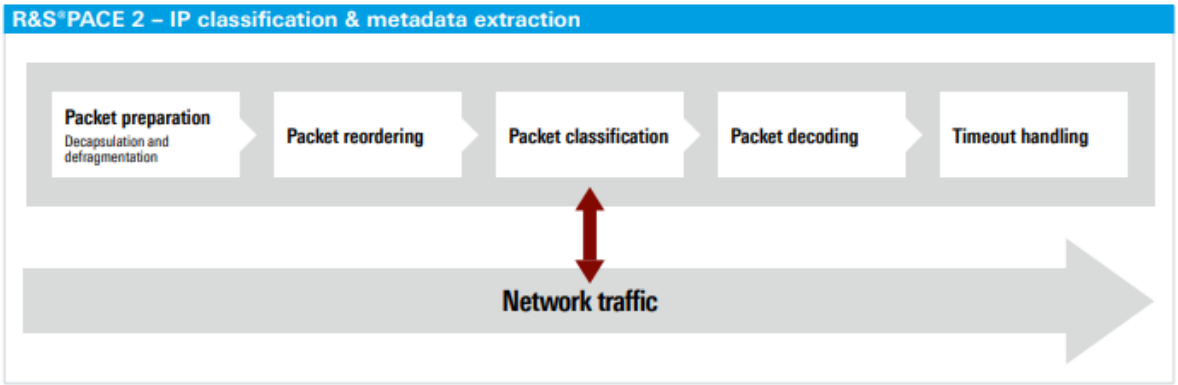

Figura 2-6 Diferentas etapas de procesamiento de un paquete en la librería R&S®PACE 2 [9].

Gracias al uso de diferentes técnicas de detección, esta librería consigue minimizar falsos positivos y negativos en la detección. Después de las pruebas realizadas por parte de la empresa del software, la marca afirma la identificación del 95% del tráfico circulante en la red. [9]

2.2.3 Firewall de nivel de aplicación.

Los firewall han evolucionado mucho en los últimos años y ya son capaces de actuar en la capa de aplicación del modelo OSI. Este tipo de firewall evalúa la capa de aplicación en los paquetes antes de permitir una conexión, también realiza un seguimiento de las conexiones así como la secuencia, con lo cual se consigue permitir tener un buen control de las conexiones e imponer un empedimento a las aplicaciones no deseadas.

Estos tipos de cortafuegos son conocidos como Proxy Firewall ya que son capaces de examinar y evaluar contraseñas, así como solicitudes de servicios que se encuentran en los datos de la capa de aplicación. Aplican por ejemplo reglas de segmentación en protocolos tales como HTTP, FTP, Telnet, etc.

También existe una variante de estos cortafuegos, denominados como firewall DPI. Estos son los más sofisticados. Pueden llegar a filtrar tipos de archivos específicos como XML ó SOAP.[10]

Aunque un firewall de aplicación proporciona bastante seguridad frente a un firewall clásico de filtrado de paquetes, el firewall de aplicación es más lento debido a que el procesamiento de los datos en el nivel de aplicación resulta ser más costoso. Esto provoca una escasa escalabilidad. [11]

ConfigServer Security Firewall (CSF) sería un ejemplo de firewall de nivel de aplicación.

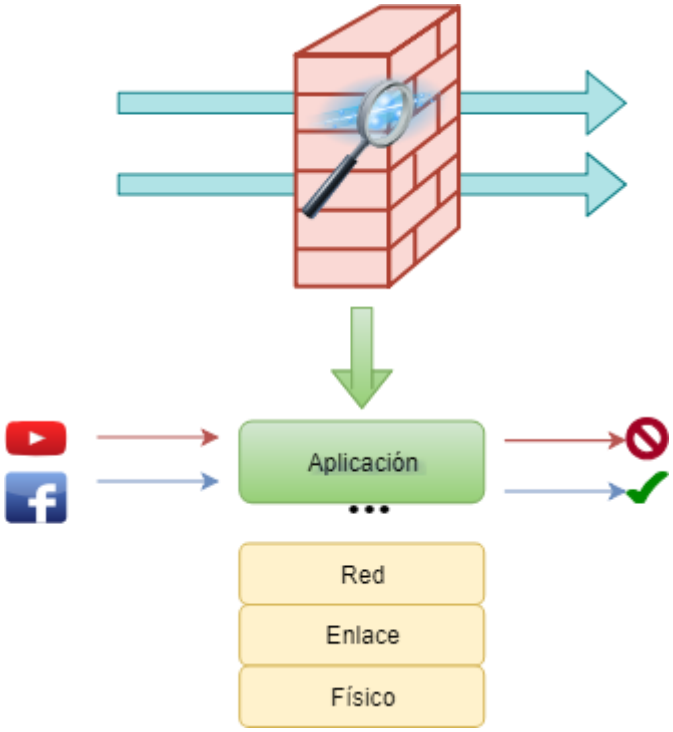

Figura 2-7 Concepto de Firewall de aplicación.

2.2.4 IDS/IPS.

La mayoría de las empresas u organizaciones desean prevenir intrusiones en su red. Los cortafuegos evitan ataques con procedencia del exterior, pero cada vez son más frecuentes ataques desde el interior de la red. Es por ello por lo que es tan importante la implantación de un sistema de detección de intrusos (IDS). Así como un sistema de prevención de intrusos (IPS) que una vez detecta y aprende un ataque, lo bloquea y genera

documentación del mismo.

Los IDS detectan ataques aplicando diferentes metodologías [12], [13]:

- Basado en firmas:
    - Realizan comparaciones de ataques con una base de datos de reglas ó firmas.
    - Posee limitaciones a la hora de la detección de nuevos ataques.
    - Generación de logs.
- Basado en anomalías:
    - Emplea un algoritmo de aprendizaje.
    - Posibilidad de detectar nuevos ataques con una probabilidad de acierto.
    - Generación de logs.

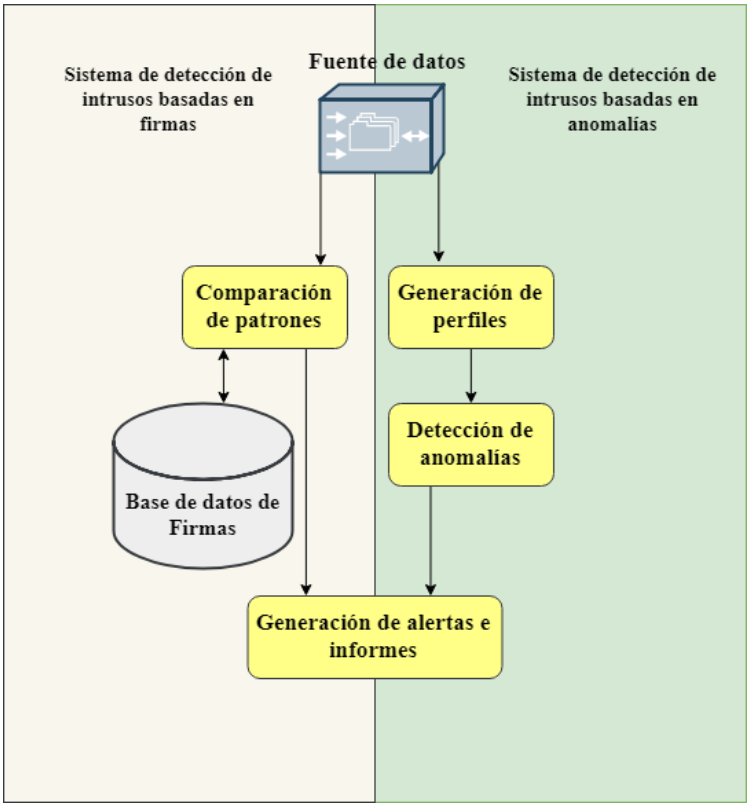

Figura 2-8 Arquitectura de IDS según la metodología.

Snort es un ejemplo claro de IDS que podemos encontrar en la actulidad. Tiene un motor de búsqueda y su metodología es basada en el análisis de firmas. Inspecciona el tráfico de red y permite la generación de alarmas. [14]

Figura 2-9 Logo de Snort

Snort funciona bajo licencia GPL, es decir, se trata de software libre. Contiene la característica de ser

multiplataforma, puesto que es posible ser ejecutado en Windows y UNIX/Linux. Así mismo, Snort lleva por defecto tanto reglas como patrones de vulnerabilidades y ataques ya conocidos e identificados.

2.2.5 Análisis y conclusiones.

Según los datos obtenidos por CVE Details [15] podemos observar un aumento de las vulnerabilidades encontradas en los últimos 5 años. Resulta interesante observar el gran aumento de vulnerabilidades que hubo en el año 2017 respecto al año anterior.

Cabe señalar que a pesar de una pequeña minoría de vulnerabilidades en el año actual a la redacción del presente documento (2018) respecto al año 2017, aún sigue existiendo bastantes. Esto se debe al gran aunmento de los diferentes dispositivos y tecnologías existentes en el mercado actual, los cuales hacen que sea de mayor envergadura el abánico de opciones para la localización y clasificación de nuevas vulnerabilidades.

Analizando estos datos cabe destacar la gran importancia de la investigación y desarrollo de nuevos sistemas para mitigar dichas brechas de seguridad. Por ello, existen un gran un número de organizaciones en los diferentes sectores tanto públicos como privados trabajando e investigando en el desarrollo de nuevas técnicas para la detección y mitigación de las nuevas vulnerabilidades.

En el siguiente gráfico se muestra una evolución temporal de las vulnerabilidades en los últimos 5 años.

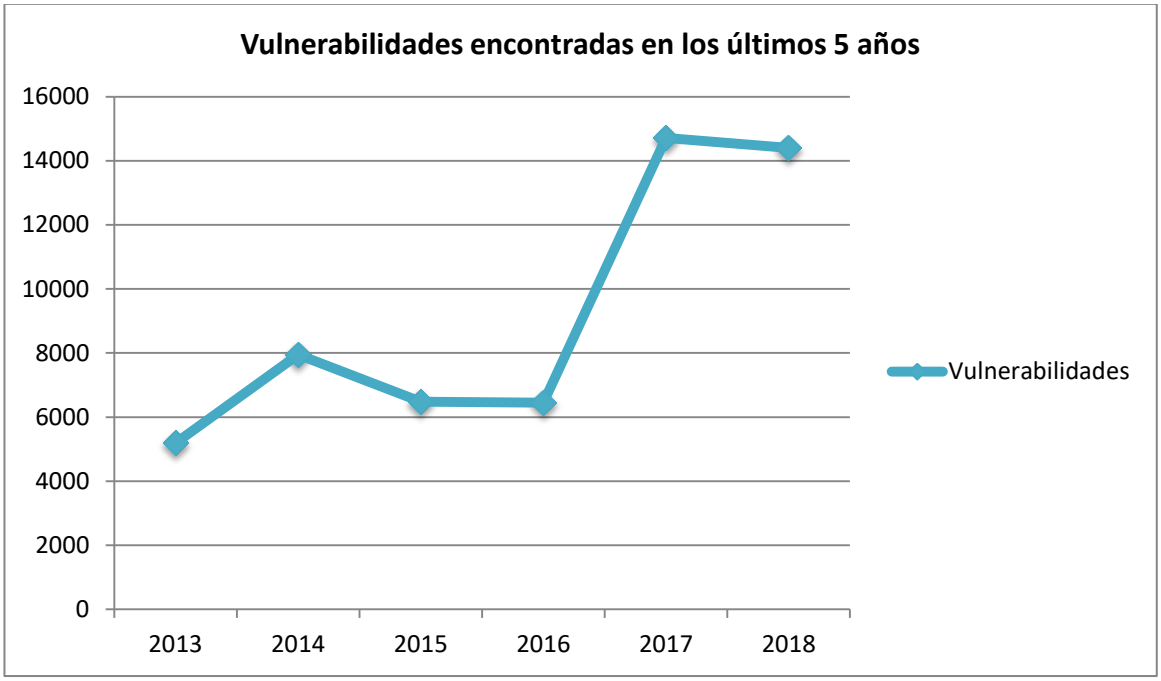

Figura 2-10 Gráfico de las vulnerabilidades en los últimos 5 años.

Según CVE Details [16] existen 13 tipos de vulnerabilidades, donde lás más comunes son las de denegaciones de servicios ó la ejecución de código malicioso.

Debido a la gran variedad de vulnerabilidades, es muy importante el desarrollo de sistemas inteligentes capaces de detectar aplicaciones y datos no deseados.

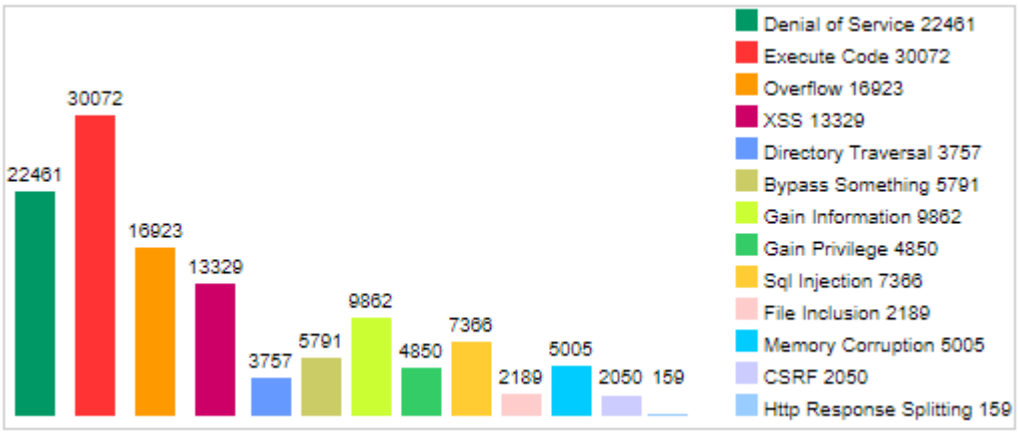

Figura 2-11 Vulnerabilidades por tipo [16].

Después de la documentación y análisis de diferentes librerías de software para la detección de aplicaciones en tráfico de red. Se puede garantizar que nDPI puede ser la mejor opción. Existe un artículo [17] donde se discuten las diferentes librerías existentes y se puede observar como nDPI resulta ser la mejor, puesto que es una librería de software libre y aún que a priori no es la librería que más aplicaciones puede detectar, se puede configurar como se desee para poder identificar protocolos y aplicaciones nuevas.

Existe una librería comercial llamada PACE que detecta numerosas aplicaciones. Tanto nDPI como PACE se basan en una antigua librería llamada OpenDPI.

Es cierto que en sus orígenes nDPI tenía números falsos positivos, pero tanto en las actualizaciones como en el desarrollo nuevas versiones esta eficencia ha sido mejorada considerablemente.

A continuación se puede observar una comparativa de las librerías más famosas usadas en la actualidad.

Nombre	Versión	Aplicaciones	Sofware Libre
PACE	2	2800	NO
nDPI	2.4	185	SI
L7-Filter [18]	(May 2009)	110	SI
NBAR2 [19]	(Jun 2013)	~1000	NO

Tabla 1 Comparativa de diferentes DPI.

Como se ha mencionado anteriormente los cortafuegos han sufrido un gran cambio en los últimos años, debido a esto resulta de gran interés realizar un análisis y comparativa de los diferentes enfoques que puede tener un firewall.

En la siguiente tabla se puede observar una comparativa de los mismos.

Funcionalidad	Firewall clásico	Firewall stateful	Firewall Aplicación	Firewall DPI
Reglas básicas	SI	SI	SI	SI

Reglas con relación entre paquetes	NO	SI	SI	SI
Reglas de aplicación	NO	NO	SI	SI
Eventos y logs	SI	SI	SI	SI

Tabla 2 Comparativa de los diferentes cortafuegos.

Para finalizar se ha realizado un estudio de dos software muy usados en la actualidad para la implementación de sistemas de detección de intrusos (IDS). Cabe destacar que ambos poseen características muy similares.

Tanto Snort como Suricata pueden implementarse para ser sitemas de prevención de intrusos (IPS) y poder cortar ataques en tiempo real.

A continuación se muestra una comparativa de los softwares mencionados.

Nombre	Enfoque	Sofware libre	Registro de eventos	Detección de aplicaciones	Versión	Multihilo
Snort	Basado en firmas	SI	SI	SI	2.9.12	NO
Suricata	Basado en firmas	SI	SI	SI	4.0.5	SI

Tabla 3 Comparativa de diferentes IDS.

3 CONCEPTOS

> You see, wire telegraph is a kind of a very, very long cat. You pull his tail in New York and his head is meowing in Los Angeles. Do you understand this? And radio operates exactly the same way: you send signals here, they receive them there. The only difference is that there is no cat.
>
> *- Albert Einstein-*

En la investigación llevada a cabo se han visto involucradas diferentes tecnologías y mecanismos. Es por ello que en este capítulo se tratan los conceptos y metodologías empleadas para el desarrollo del proyecto.

3.1 IPFIX y Netflow.

IPFIX proviene de las siglas en inglés *IP Flow Information Export* [20], el cuál especifica un mecanismo estándar de exportación de datos sobre el flujo de red, tanto en conmutadores como en routers.

Se puede definir como flujo de red un conjunto de paquetes IP pertenecientes a una misma conexión.

La compañía CISCO Systems desarrolló en 1996 una tecnología con el fin de mejorar el encamienamiento de los routers. Esta tecnología denominada Netflow, se basa en la identificación de los flujos establecidos entre distitos dispositivos para así poder agilizar el encaminamiento de paquetes IP. [21]

Un flujo de datos se caracteriza por la combinación de diversos atributos en un intervalo de tiempo. Estos atributos suelen ser direcciones, puertos, protocolo identificado, servicio, así como la interfaz de entrada.

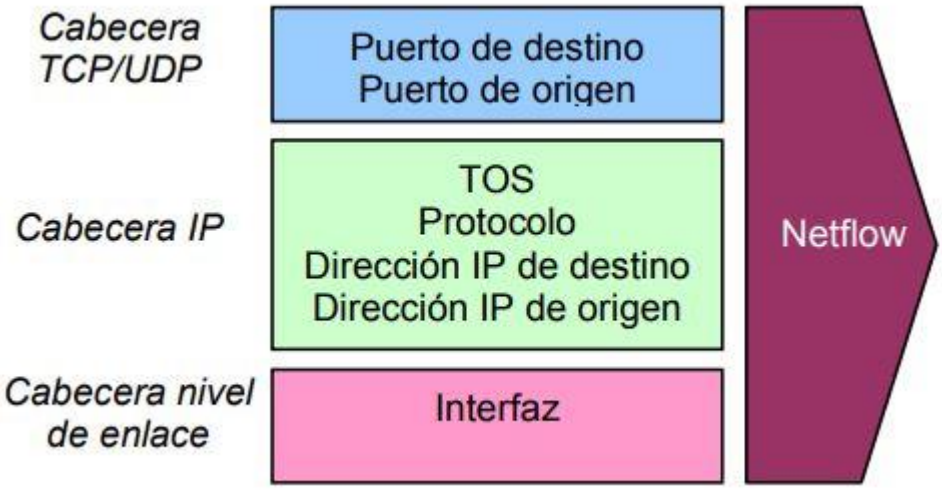

Figura 3-1 Información exportada a través de Netflow.[21]

Cuando un router identifica un nuevo flujo, Netflow lo guarda junto a la interfaz de salida asociada al flujo, esto hace que no sea necesario realizar consultas de encaminamiento para paquetes posteriores correspondientes al flujo. Consiguiendo disminuir el coste computacional que implica las consultas a las tablas de encaminamiento.

Una gran ventaja que provee esta carácterística, es la posibilidad de poder realizar mediciones y caracterizaciones del tráfico cursante en tiempo real.

Netflow es usado comumente para realizar análisis de la calidad del servicio a través de definiciones de métricas.

3.2 DPI.

Las primeras técnicas usadas para la identificación y clasificación de flujos de red estaban basadas en su mayoría, en el uso de puertos conocidos.[22] Es decir, se analizaban los encabezados de los paquetes para identificar un servicio en base al puerto. Por ejemplo, si llegaba tráfico asociado al puerto 80, dicho tráfico sería clasificado e identificado como HTTP. Este comportamiento presenta debilidades debido a que es impreciso. El tráfico cursante podría ir tunelado [23] consiguiendo así evitar reglas de un firewall.

Figura 3-2 Tráfico tunelado.

Según un artículo [22] menos del 30% del tráfico es clasificado correctamente haciendo uso de identificación basada en puertos. Debido a esto, los sistemas han reemplazado la metodología basada en puertos por la inspección profunda de paquetes, consiguiendo así un gran aumento en la precisión.

DPI consigue analizar la carga útil de un paquete y encontrar patrones que se identifiquen con un servicio.

Cabe destacar, que este tratamiento puede llegar a ser muy costoso, por lo que los patrones son identificados por expresiones regulares, ya que así se consigue disminuir el coste computacional.

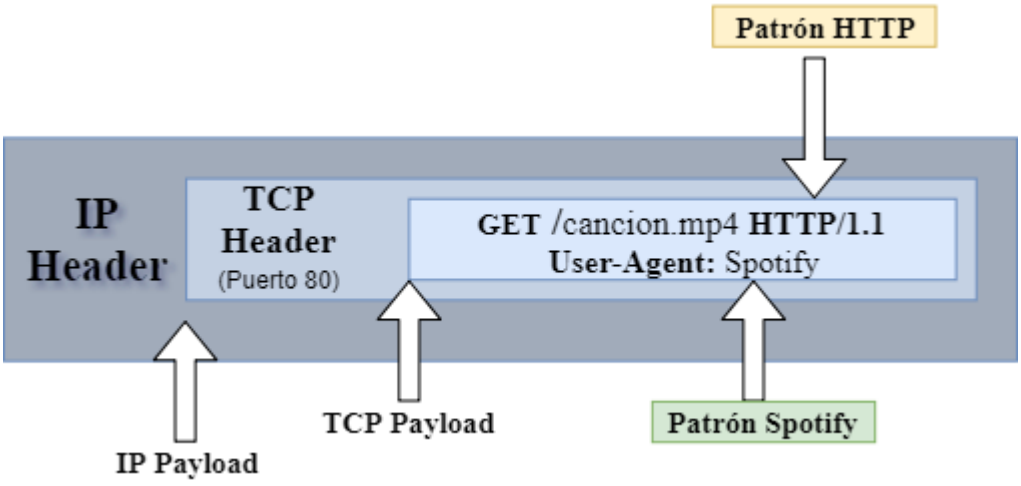

Figura 3-3 Identificación de patrones usando DPI.

Con este tratamiento los ISP [24] pueden proporcionar calidad de servicio a una aplicación en concreto, ó realizar cobros según el servicio que se esté utilizando.

3.3 nDPI.

Como ya se mencionó anteriormente nDPI es una librería capaz de realizar inspección profunda de paquetes, programada en **lenguaje C**.

Esta librería es multiplataforma, puesto que puede ser usada tanto en UNIX como en Windows.

Aunque por defecto es capaz de detectar numerosas aplicaciones, es muy fácil añadir nuevos servicios que no se puedan identificar a priori.

Consultando el manual [25] podemos observar un ejemplo de como añadir nuevos protocolos para que sean identificados en el flujo de red.

```
# Subprotocols
# Format:
# host:"<value>",host:"<value>",.....@<subproto>
host:"googlesyndacation.com"@Google
host:"venere.com"@Veneer
```

Figura 3-4 Añadir protocolos en nDPI.

Debido a que acualmente la mayoría de las conexiones se encuentran cifradas, nDPI contiene un decodificador SSL [26], el cual es capaz de extraer el nombre del host del certificado perteneciente al servidor. Esta información se añade a los metadatos del flujo de red, consiguiendo así ser identificado.

La librería nDPI implementa el algoritmo de ***Aho-Corasick*** para encontrar patrones dentro de un texto. Con el fin de ser eficiente a la hora de analizar la carga últil de los paquetes circulantes en la red.

Este algoritmo implementa un autómata similar a una estructura de datos de tipo árbol. A continuación se muestra un ejemplo del algoritmo cuando los patrones son: {***a, ab, bc, bca, c, caa}***

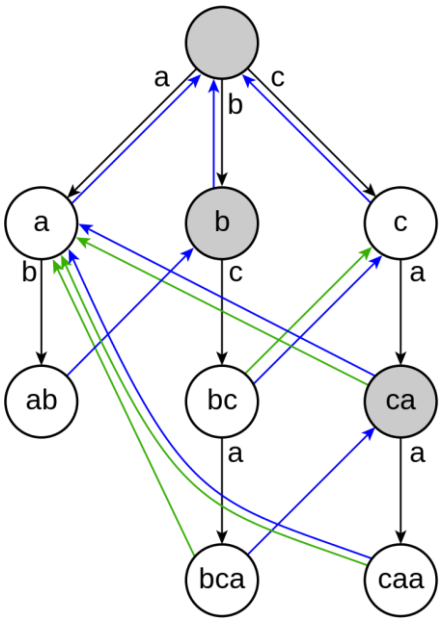

Figura 3-5 Ejemplo de algoritmo Aho-Corasick. [27]

La característica más interesante que posee este algoritmo, resulta ser la capacidad de poder bucar patrones de forma simultánea.

Esta implementación es usada mucho en el mundo de la ciberseguridad, como es por ejemplo para las bases de datos de virus.

3.4 wireshark.

Wireshark es un software usado para analizar protocolos y pasos de mensajes. Esta aplicación permite capturar todos los paquetes de una red con el fin de poder identificar problemas de configuración en las redes de comunicación.

Figura 3-6 Logo de wireshark.

Esta aplicación ha tenido gran importancia en el estudio realizado, puesto que ha sido utilizada para capturar el tráfico que será usado como entrada al sistema desarrollado.

3.5 Integración de nDPI con wireshark.

Wireshark posee la posibilidad de ser integrada junto con la librería nDPI en los sistemas UNIX. Esta nueva funcionalidad fue presentada en el SharkFest de 2017 [28]. Y permite la identificación de las aplicaciones en los paquetes capturados.

Para hacer esto posible, es necesario descargar de github los ficheros ***ndpiReader.c*** junto a ***ndpi.lua.***

A continuación se muestra un ejemplo donde se ha capturado tráfico de **Google** y se puede comprobar cómo se ha añadido la información en el campo protocolo. Además, también se puede realizar filtros para mostrar sólo cierta aplicación. Un ejemplo sería: ***`ndpi.protocol.name == HTTP.Google`***

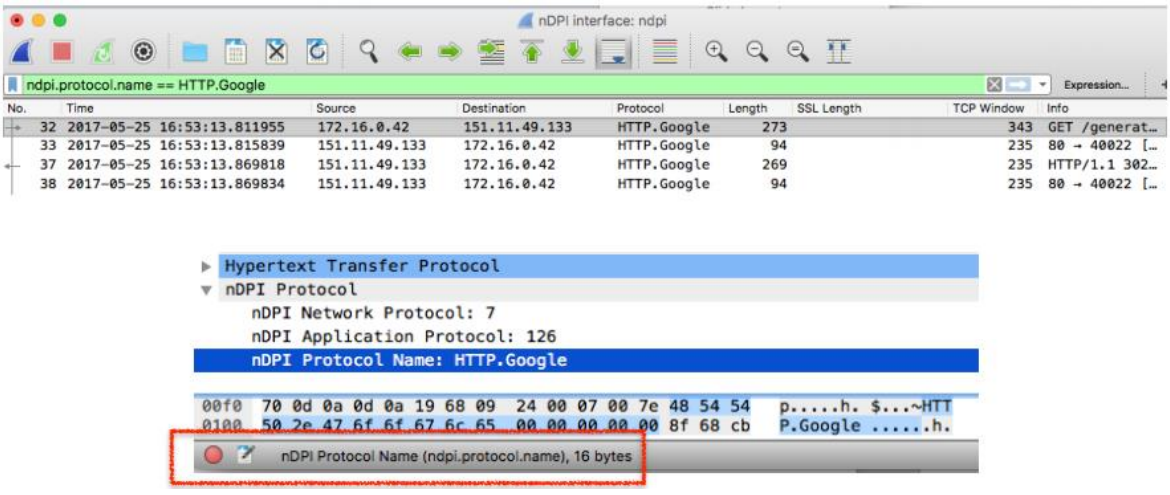

Figura 3-7 Captura de paquetes con wirehsark y nDPI.

Una de las ventajas de poder capturar identificando las aplicaciones, es la de poder capturar sólo los paquetes de ciertas aplicaciones. Esto permite una mejora en la capacidad de respuesta de wireshark reduciendo el uso de recursos.

Este plugin también es capaz de llevar a cabo un seguimiento de los certificados del servidor, así como el conteo de números de conexiones SSL por certificado.

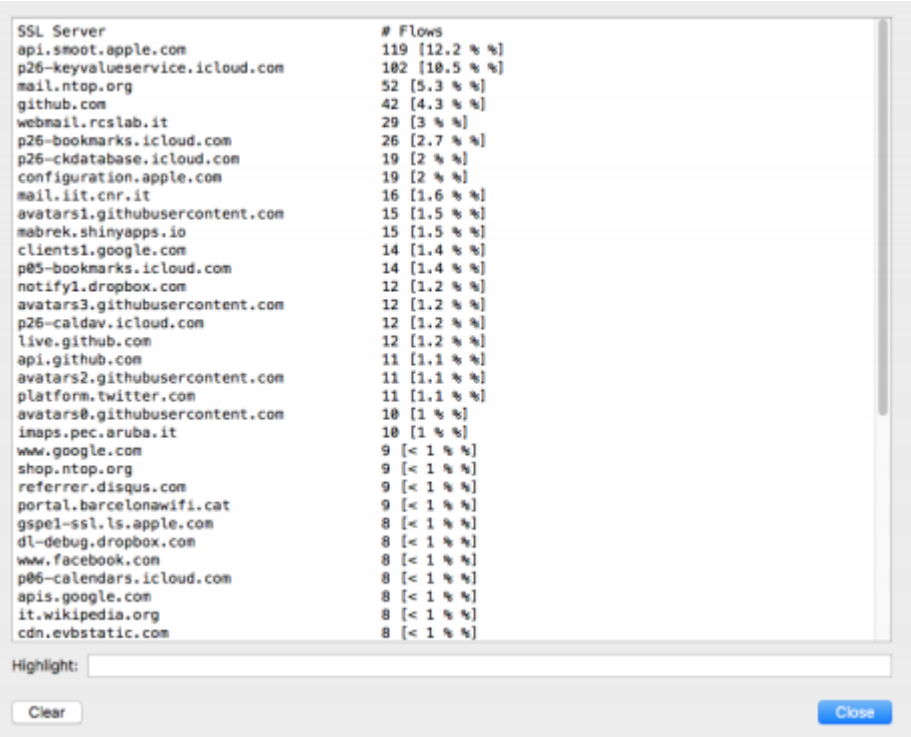

Figura 3-8 certificados SSL.

3.6 Recolector nProbe.

nProbe es una sonda NetFlow v5/v9 IPFIX que recompila, analiza y exporta información del tráfico de red. Este software posee muchas funcionalidades, pero en el estudio llevado a cabo se ha usado para poder exportar los flujos del tráfico capturado en formato IPFIX. Se puede encontrar una gran información sobre este software en el manual [29].

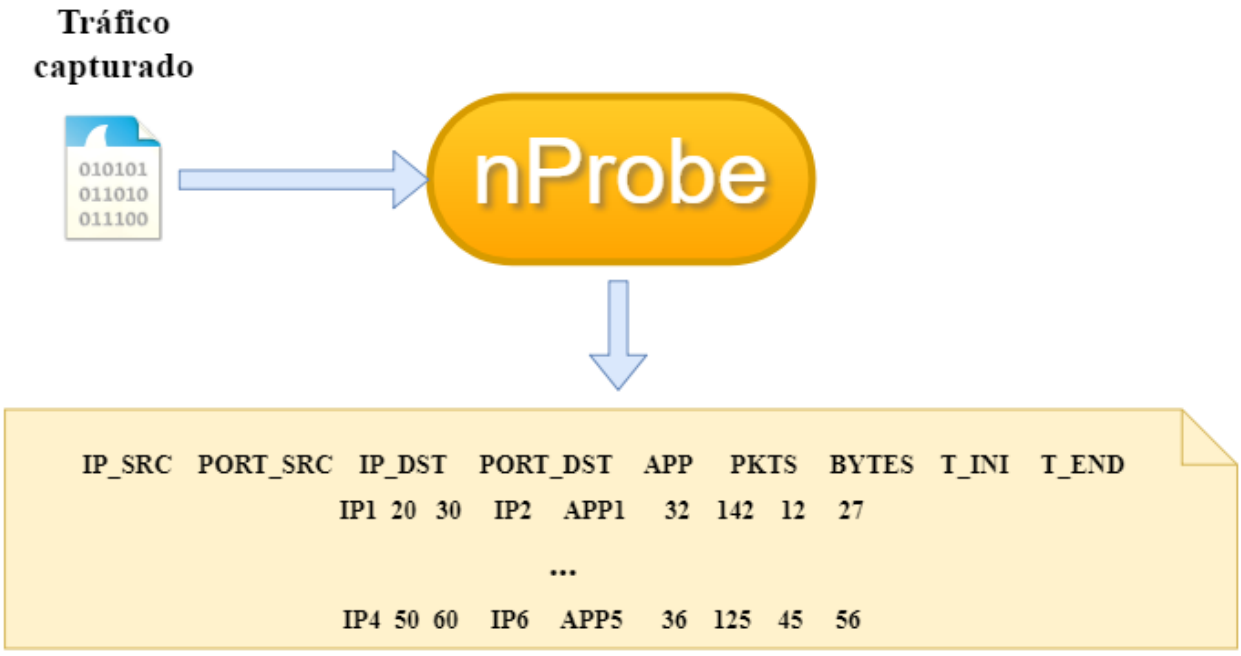

Figura 3-9 Uso de nProbe.

La gran ventaja del uso de nProbe es la posibilidad de exportar la información que desee. Además nProbe integra internamente la librería nDPI, por lo que se puede añadir la información de que aplicación corresponde al flujo.

4 SISTEMA

My work on free software is motivated by an idealistic goal: spreading freedom and cooperation. I want to encourage free software to spread, replacing proprietary software that forbids cooperation, and thus make our society better.

- Richard Stallman -

En este capítulo se verá el diseño del sistema desarrollado, las partes en las que se compone, así como las principales funcionalidades que posee. Cabe destacar que el sistema podría ser usado en casos reales debido a la implementación realizada.

4.1 Aspectos generales del sistema.

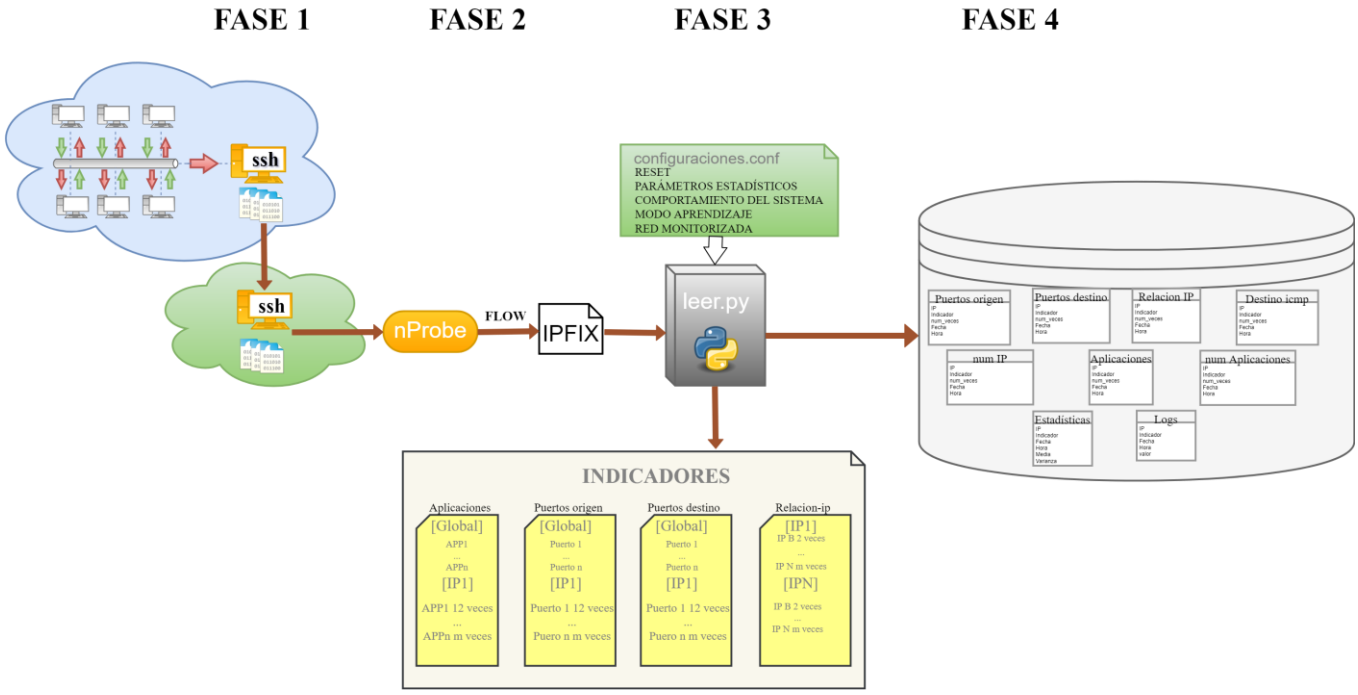

Figura 4-1 Esquema general del sistema.

Debido a la complejidad que posee el sistema desarrollado, para una mejor compresión del mismo, este puede ser descompuesto en 4 fases:

- Una primera en la que se capturan los datos.
- Una segunda parte en la que se exporta la información en formato IPFIX.
- Una tercera encargada del procesamiento de todos los datos.
- Y por último la exportación final del tratamiento de los datos.

4.2 Fase 1: Obtención del tráfico.

Primero será necesario capturar todo el tráfico que circula en una red. Para ello se hace uso de tcpdump[30] en modo promiscuo.

Una vez capturados los paquetes estos son guardados en un archivo de tipo ***pcap.***

El equipo encargado de realizar las capturas, guardará un archivo ***pcap*** por cada franja horaria, consiguiendo así realizar un sondeo de la red a cada hora.

Estos archivos son enviados a un segundo equipo ubicado en una red externa, puesto que si procesaramos la información en el mismo equipo, podríamos sobrecargarlo y no conseguir el fin deseado.

Es muy importante que para el envío de los archivos de capturas, se realice a través de un medio seguro como es por ejemplo **SCP** [31]. El cuál permite transferencias seguras de ficheros a través de **SSH** [32].

4.3 Fase 2: Exportación del tráfico en formato IPFIX.

Hasta ahora la información del tráfico circulante en la red se encuentra en bruto en archivos ***PCAP.*** En esta fase se usa como entrada a nProbe estos ficheros, con el fin de poder exportar la información en archivos ***CSV*** [33] **.**

La información exportada representa las comunicaciones en la red a modo de flujos IP en formato IPFIX. A continuación se muestra un ejemplo:

```
IPV4_SRC_ADDR,L4_SRC_PORT,IPV4_DST_ADDR,L4_DST_PORT,L7_PROTO,L7_PROTO_NAME,IN_PKTS,IN_BYTES,FLOW_START_MILLISECONDS,FLOW_END_MILLISECONDS
91.228.166.150,443,192.168.0.26,2594,188,QUIC,5,637,1537728302979,1537728302979
192.168.0.26,2594,91.228.166.150,443,188,QUIC,8,3448,1537728302979,1537728302979
192.168.0.26,2595,91.228.166.150,443,91,SSL,16,3837,1537728302979,1537728302980
91.228.166.150,443,192.168.0.26,2595,91,SSL,10,4355,1537728302979,1537728302980
192.168.0.26,63337,239.255.255.250,1900,12,SSDP,129,21285,1537728302980,1537728303080
192.168.0.26,2596,91.228.166.150,443,91,SSL,16,4168,1537728302980,1537728302980
91.228.166.150,443,192.168.0.26,2596,91,SSL,11,4391,1537728302980,1537728302980
192.168.0.26,55478,172.217.16.238,443,188,QUIC,1,51,1537728302980,1537728302980
172.217.16.238,443,192.168.0.26,55478,188,QUIC,1,48,1537728302980,1537728302980
192.168.0.26,1837,92.122.188.120,443,188,QUIC,5874,392245,1537728302980,1537728303080
92.122.188.120,443,192.168.0.26,1837,188,QUIC,24710,36184858,1537728302980,1537728303080
192.168.0.26,2515,35.186.224.53,443,188,QUIC,5,203,1537728302980,1537728303002
35.186.224.53,443,192.168.0.26,2515,188,QUIC,6,339,1537728302980,1537728303002
```

Figura 4-2 Ejemplo de exportación con nProbe.

A continuación se muestra una tabla donde se describe la información que se va a exportar:

Campo	Descripción
• IP origen	IP versión 4 que identifica al origen del flujo IP.
• IP destino	IP versión 4 que identifica al destino del flujo IP.
• Puerto origen	Puerto origen de capa 4 usado para el transporte del flujo.
• Puerto Destino	Puerto destino de capa 4 usado para el transporte del flujo.

• Nº Aplicación	Número del identificador de la aplicación que ha sido encontrada en el tratamiento DPI.
• Nº de paquetes	Cantidad de paquetes totales que corresponden al flujo.
• Nº de Bytes	Cantidad de Bytes totales en el flujo.
• Instante inicial	Tiempo de inicio del flujo.
• Instante final	Tiempo final del flujo.

Tabla 4 Campos exportados con nProbe.

Como ya se mencionó anteriormente, nProbe tiene muchas funcionalidades y es muy versátil a la hora de exportar la información. Debido a que este estudio solo requiere de los campos mostrandos en la *Tabla 4*, es necesario ejecutar nProbe como se muestra en la imagen *4-3*.

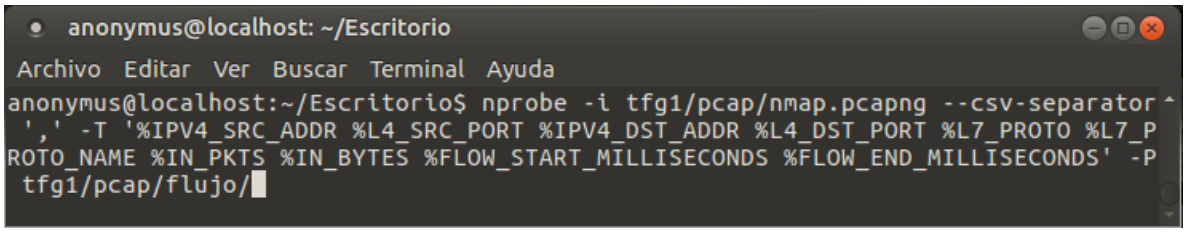

Figura 4-3 Ejecución nProbe.

En la *tabla 5* se detallan los parámetros usados en nProbe.

Opción	Descripción
• -i	Indica la interfaz o fichero PCAP de entrada.
• --csv-separator	Impone que la salida del fichero tenga formato csv.
• -T	Esta opción permite el orden y la información que se desea exportar.
• -P	Selección la carpeta donde se desea guardar la información a exportar.

Tabla 5 Descripción de las opciones usadas en nProbe.

Una vez se realiza este procesamiento. En la carpeta destino especificada con el parámetro *–P* aparecerá un fichero de tipo *.flows,* el cuál tendrá la información deseada.

4.4 Fase 3: Procesamiento de IPFIX.

Esta etapa es la más importante, debido a que resulta ser el peso grande del estudio realizado. Hasta ahora los datos no se han analizado para detectar anomalías en el tráfico de red. Es en esta fase donde se realiza el tratamiento de la información para toma de decisiones en la detección de comportamientos anómalos.

4.4.1 Lenguaje de programación y librerías.

El lenguaje de programación utilizado para el desarrollo del trabajo ha sido **Python** [34], puesto que es el lenguaje más extendido y usado en el mundo de la ciberseguridad.

Figura 4-4 Logo de Python [34]**.**

Además, este lenguaje tiene la posibilidad de usar numerosas librerías muy útiles que facilitan el trabajo del programador. En la *tabla 6* podemos observar las principales librerías utilizadas para el desarrollo.

Librería	Descripción
• Pandas [35]	Esta librería permite el análisis de datos de forma eficiente. Ofrece diferentes estructuras de datos como son: • Data Frame. • Series. • Estructuras de datos multidimensionales.
• Numpy [36]	Destinada para poder realizar operaciones con matrices y vectores.
• Datetime [37]	Módulo que proporciona clases y objetos para la manipulación de horas y fechas.
• Os [38]	Permite ejecutar comandos propios del sistema operativo.
• Sys [39]	Proporciona variables y funcionalidades con el propio intérprete de Python.
• mysql.connector [40]	Permite la interacción con una base de datos MySQL.
• math [41]	Dota a Python de operaciones matemáticas.
• shutil [42]	Proporciona acceso y manipulación de documentos y directorios alojados en el sistema operativo.
• ConfigParser [43]	Permite el acceso de forma sencilla a ficheros de configuración.
• Ipcalc [44]	Permite realizar cálculos de subredes IP tanto en IPv4 como IPv6.

Tabla 6 Librerías utilizadas.

4.4.2 Diseño del sistema.

En la figura *4-5* se muestra el diseño llevado a cabo para la implentación del sistema desarrollado.

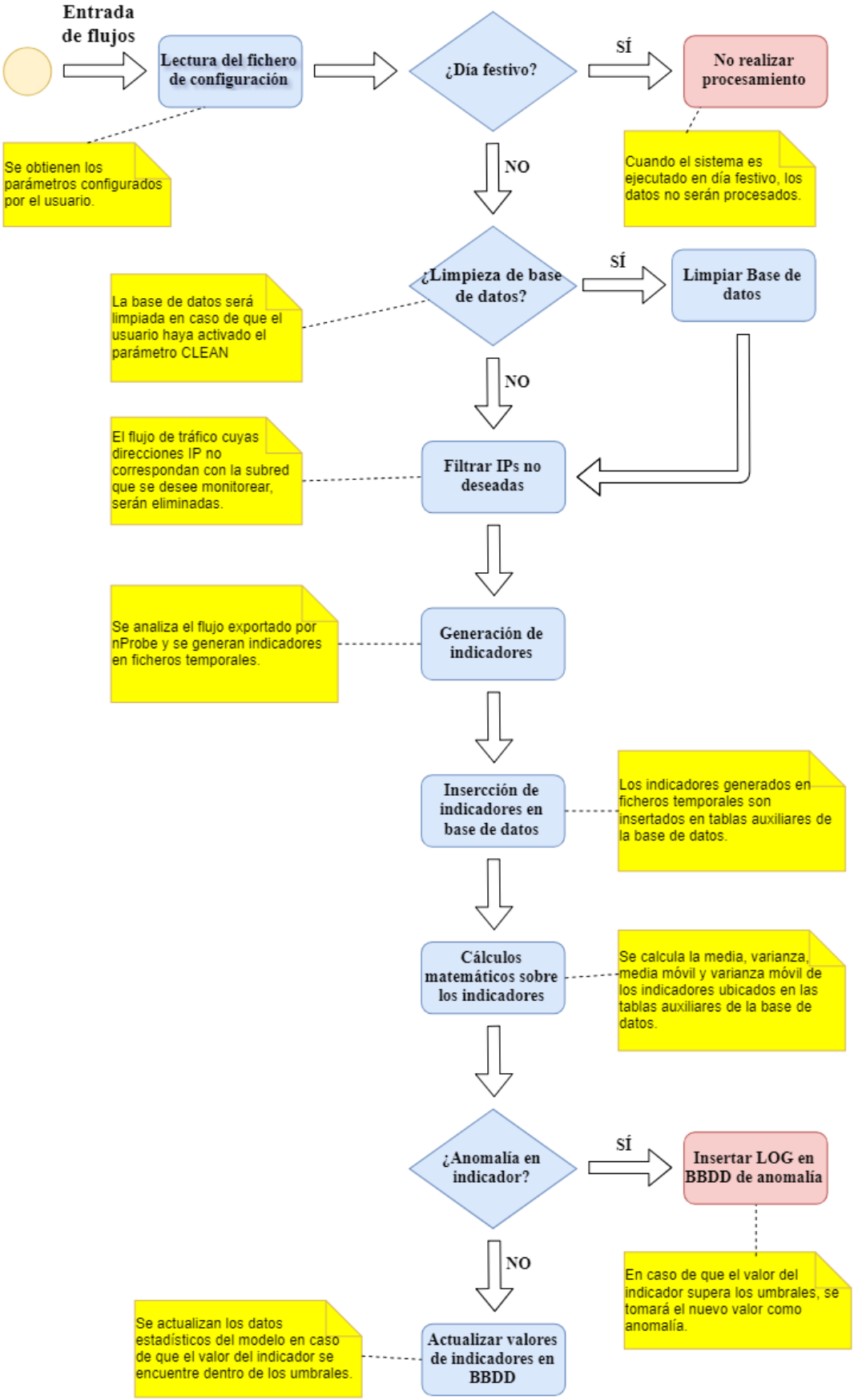

Figura 4-5 Diseño del sistema.

4.4.2.1 Diseño de los principals componentes del sistema.

Lectura del fichero de configuración	
Descripción:	Obtención de los parámetros definidos por el usuario. Para facilitar al usuario, los parámetros se encuentran divididos en 4 secciones: • FECHA • BBDD • ESTADISTICAS • TRATAMIENTO
Parámetro	**Uso y opciones**
Fecha	Define la fecha y hora que se desee asignar a los indicadores. Este parámetro por defecto es AUTO, en este valor la fecha y hora se asignan automáticamente. En caso de estar realizando pruebas o desarrollando, se puede asignar manualmente en el formato YYYYMMDDHH. Un ejemplo sería: 2018092117, que correspondería con el 21 de septiembre de 2018 a las 17:00.
User	Define el usuario miembro de la base de datos.
Password	Define la contraseña de la base de datos.
Host	Define la dirección IP donde se encuentra alojada la base de datos.
Database	Define el usuario miembro de la base de datos.
Clean	Si se desea limpiar la base de datos este valor deberá estar a TRUE, en caso contrario deberá estar a FALSE.
Capacidad_ventana	Define el tamaño de las N últimas muestras que se tienen en cuenta en los cálculos matemáticos realizados.
Coeficiente_error	Coeficiente de error para la toma de decisión. Normalmente este valor estará comprendido entre 2 y 4. Cuanto más pequeño sea este valor el sisteme será más restrictivo.
Aprendizaje	Este parámetro es muy importante, ya que cuando su valor es igual a TRUE, el sistema se encuentra en modo de aprendizaje. Es decir, el sistema se encuentra en modo de entrenamiento. Todos los datos de entrada serán clasificados como tráfico limpio y valdrán para realizar una modelización del tráfico. Cuando el parámetro se encuentra a FALSE el sistema evaluará todos los indicadores, generando alertas en caso de detectar anomalías.

Eliminar_anomalos	Si el parámetro se encuentra a TRUE, los valores que provoquen una alarma no se insertarán en la base de datos. Si por lo contrario, se desea que estos valores se inserten en base de datos, el parámetro deberá estar a FALSE.
Alertas_nuevas	Si se encuentra a TRUE y se detecta un nuevo indicador, el sistema generará una alarma notificando el nuevo indicador. Si no se desea alertar de nuevos indicadores, el valor deberá esta a FALSE.
Alpha	Parámetro de suavizado de los cálculos estadísticos, este valor suele estar comprendido entre 0.1 y 0.5
network	Define el rango de subred de direcciones IP que se desean tener en cuenta para el estudio. Un ejemplo de valor sería: 192.168.0.1/24 dónde se estarían indicando todas las IPs de la subred 192.168.0.1 con máscara 24.

Tabla 7 Parámetros del fichero de configuración.

Filtrar IPs no deseadas
La información exportada en formato IPFIX puede contener flujos de direcciones IPs que no se desean tener en cuenta para el estudio. Es por ello por lo que las direcciones IP origen correspondientes a la subred definida en el parámetro ***network*** serán convertidas a la dirección IP 1.1.1.1 y las direcciones IP destino a la 2.2.2.2. Este tratamiento se realiza para eliminar el tráfico de direcciones irrevelantes para el estudio así como para facilitar y simplificar el procesamiento de los datos.

Tabla 8 Tratamiento de las direcciones IP del tráfico.

Generación de indicadores	
En base al flujo IPFIX filtrado se crean los siguientes indicadores:	
Indicador	**Descripción**
Aplicaciones	Recoge las diferentes aplicaciones detectadas en los flujos así como el número de veces que intervienen en ellos.
ICMP-destino	Muestra el número de veces que se han generados mensajes ICMP[45]. Este indicador puede alertar de ataques de denegación de servicios.
IPdestino-origen	Informa por cada IP destino el número de veces que se conecta con las diferentes IP origen.
IPorigen-destino	Informa por cada IP origen el número de veces que se conecta con las diferentes IP destino.
Numero-APP	Número de veces que aparace cada aplicación en el tráfico de la red.
Numero-IP	Número de IPs destino distintas.
PuertosDestino	Registro de todos los puertos destino por cada IP origen.
PuertosOrigen	Registro de todos los puertos origen por cada IP origen.

Tabla 9 Diferentes indicadores.

4.4.3 Cálculo de los indicadores.

Para crear un modelo estadístico se parten de dos tipos de premisas:

1. Mediante cálculo de la media aritmética móvil[46] y varianza[47]:

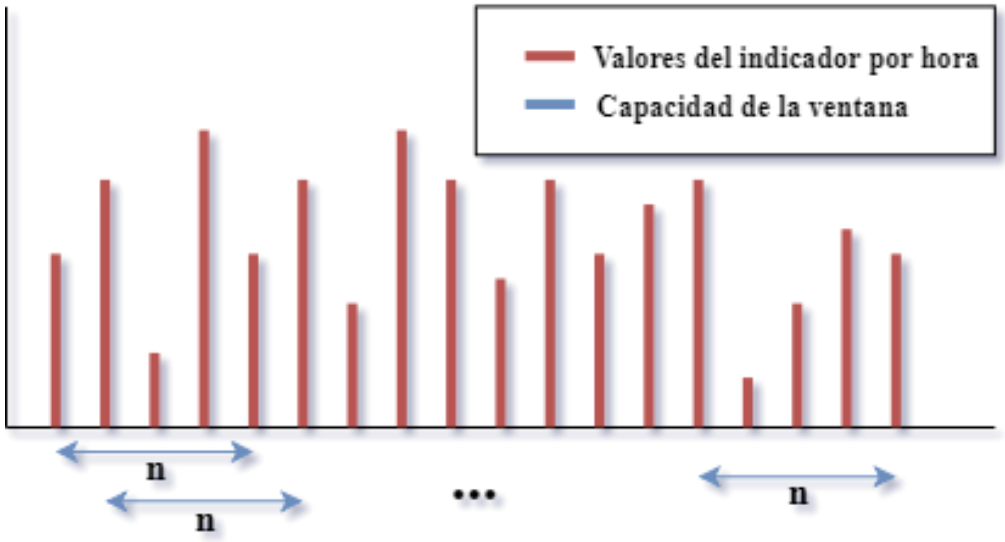

Figura 4-6 Media móvil.

- Como se puede observar en la figura ***4-7,*** el cálculo realizado corresponde a una media móvil, debido a que se calcula el promedio de las n últimas muestras. Este comportamiento se denomina como ventana deslizante. Se ha implementado de esta forma para conseguir que el sistema pueda responder a lo largo del tiempo. Ya que si no se implenta una ventana deslizante, llegaría un punto en el que la media no varía y su resultado sería prácticamente constante.
- La media aritmética o promedio cuya fórmula es:

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

  - **n** Número de la capacidad de la ventana.
  - x_i Se denota como los valores que toma la variable.

- La varianza es una medida de dispersión y la raíz cuadrada de la varianza nos permite calcular la desviación que posee cada indicador. A continuación se muestra la fórmula de la varianza:

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{X})^2$$

  - **n** Número de la capacidad de la ventana.
  - x_i Se denota como los valores que toma la variable.

- $\bar{X}$ Media de la variable.

- Cuando se almacena un valor en un indicador, este es evaluado a través de la siguiente ecuación y si no cumple la condición el valor será clasificado como anómalo:

$$\epsilon \leq k \times \sqrt{\sigma^2}$$

- ϵ Se define como el error y se calcula de la siguiente manera:

$$\epsilon = |x_i - \bar{X}|$$

- k Corresponde con el coeficiente de error. (Este parámetro es configurable por el usuario a través del fichero de configuración).

2. Mediante cálculo de la media exponencial móvil y varianza exponencial móvil:
   - La media móvil exponencial[48] cuya fórmula es:

$$EMA(t) = \begin{cases} x_1, & t = 1 \\ \alpha \times x_i + (1 - \alpha) \times EMA(t-1), & t > 1 \end{cases}$$

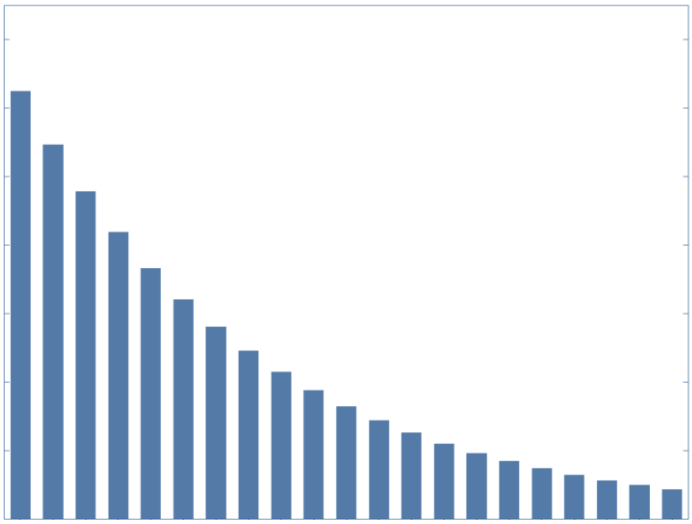

Figura 4-8 Forma de media exponencial móvil.

- α Corresponde al parámetro de suavizado que puede valer entre 0 y 1.
- x_i Valor en el instante t.
- La EMA es usado comúnmente para las tendencias de los valores de mercado. Se ha optado por implementar este módelo estadístico debido a los buenos resulados que se obtienen cuando se evalúan indicadores.

   - La varianza exponencial cuya fórmula es:

$$\sigma^2{}_{exp} = \left(\frac{\alpha}{2 - \alpha}\right) \times \sigma^2$$

- α Parámetro de suavizado.
- σ^2 Valor de la varianza de todo el conjunto de muestras.

- Al igual que en el caso anterior de media aritmética, en este también se toma decisiones en base a la siguiente fórmula:

$$\epsilon \leq k \times \sqrt{\sigma^2{}_{exp}}$$

  - $\boldsymbol{\epsilon}$ Se define como el error y se calcula de la siguiente manera:

$$\epsilon = |x_i - EMA(t)|$$

  - $\boldsymbol{k}$ Corresponde con el coeficiente de error. (Este parámetro es configurable por el usuario a través del fichero de configuración).

4.5 Fase 4: Base de datos.

Esta última etapa del sistema corresponde con la exportación a una base de datos toda la información recolectada y procesada.

Se ha utilizado MySQL[49] como sistema de gestor para la base de datos.

4.5.1 Diseño de la base de datos.

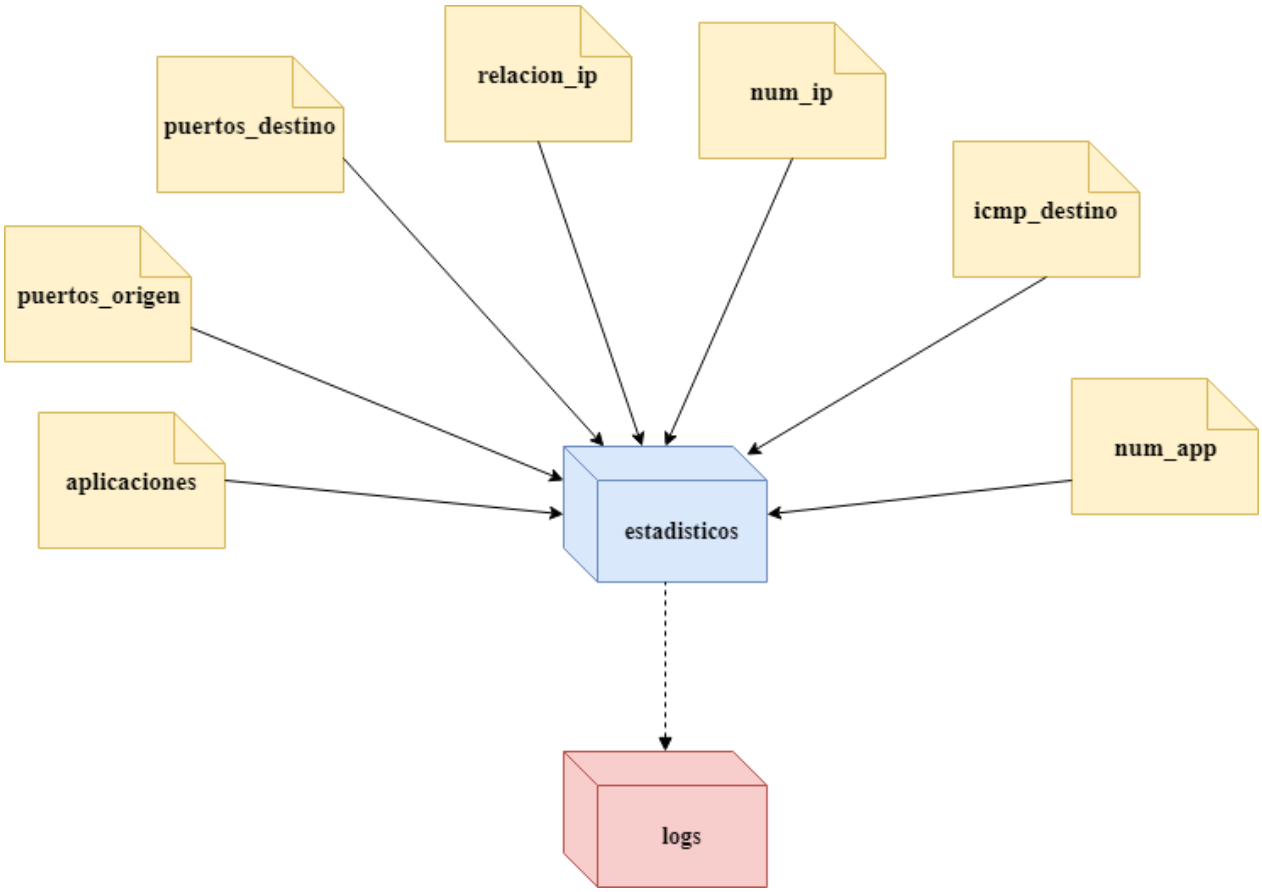

Figura 4-9 Diagrama funcional de la base de datos.

La base de datos esta compuesta por tablas auxiliares, cada una de estas tablas contiene la información de un indicador. Por cada uno de ellos se realiza un cálculo matemático y el resultado se inserta en la tabla estadísticos.

```
mysql> show tables;
+-----------------+
| Tables_in_tfg   |
+-----------------+
| aplicaciones    |
| estadisticos    |
| icmp_destino    |
| logs            |
| num_app         |
| num_ip          |
| puertos_destino |
| puertos_origen  |
| relacion_ip     |
+-----------------+
9 rows in set (0,00 sec)
```

Figura 4-10 Tablas que componen la base da datos.

Cada vez que se realiza un cálculo se comprueba que el resultado está dentro del margen de confianza. En caso de no cumplir dicho margen el nuevo valor del indicador no será insertado en la tabla **estadísticos,** pero sí se insertará un registro en la tabla de logs.

Si por lo contrario el valor si se encuentra dentro del margen de confianza, dicho valor si será insertado en la

tabla de estadísticos.

4.5.1.1 Tablas auxiliares.

A continuación se detalla el diseño de cada una de las tablas auxiliares que alimentan a la tabla principal que contienen las estadíasticas de cada indicador.

Aplicaciones	
Descripción:	Tabla que contiene el indicador de aplicaciones.
Clave primaria:	• ip • aplicacion • num_veces • fecha • hora
Atributo	**Descripción**
ip	Ip versión 4 origen del indicador.
aplicacion	Nombre de la aplicación identificada en un flujo.
num_veces	Número de flujos en los que se ha identificado la aplicación.
fecha	Fecha en formato YYYY-MM-DD al que corresponde el indicador.
hora	Hora en formato HH al que corresponde el indicador.

Tabla 10 Indicador de aplicaciones.

Icmp_destino	
Descripción:	Tabla que contiene el indicador de mensajes ICMP originados en una máquina.
Clave primaria:	• ip • ip_destino • num_veces • fecha • hora
Atributo	**Descripción**
ip	Ip versión 4 origen del indicador.
ip_destino	Ip destino a la es destinado el mensaje ICMP.

num_veces	Número de veces que se ha generado el mensaje ICMP con estas direcciones ip.
fecha	Fecha en formato YYYY-MM-DD al que corresponde el indicador.
hora	Hora en formato HH al que corresponde el indicador.

Tabla 11 Indicador de ICMP.

Puertos_destino	
Descripción:	Tabla que contiene el indicador de los puertos destino.
Clave primaria:	• ip • puerto • num_veces • fecha • hora
Atributo	**Descripción**
ip	Ip versión 4 origen del indicador.
puerto	Puerto destino identificado en el fujo.
num_veces	Número de veces que se identifica el puerto destino con la misma ip origen.
fecha	Fecha en formato YYYY-MM-DD al que corresponde el indicador.
hora	Hora en formato HH al que corresponde el indicador.

Tabla 12 Indicador Puertos destino.

Puertos_origen	
Descripción:	Tabla que contiene el indicador de los puertos origen.
Clave primaria:	• ip • puerto • num_veces • fecha • hora

Atributo	Descripción
ip	Ip versión 4 origen del indicador.
puerto	Puerto origen identificado en el fujo.
num_veces	Número de veces que se identifica el puerto origen con la misma ip origen.
fecha	Fecha en formato YYYY-MM-DD al que corresponde el indicador.
hora	Hora en formato HH al que corresponde el indicador.

Tabla 13 Indicador Puertos destino.

Relación_ip	
Descripción:	Tabla que contiene el indicador de las diferentes ip.
Clave primaria:	• ip • puerto • ip_destino • fecha • hora
Atributo	**Descripción**
ip	Ip versión 4 origen del indicador.
ip_destino	Ip versión 4 destino en el flujo.
num_veces	Número de veces que ha aparecido la ip destino para la misma ip origen.
fecha	Fecha en formato YYYY-MM-DD al que corresponde el indicador.
hora	Hora en formato HH al que corresponde el indicador.

Tabla 14 Indicador relación de las diferentes ip.

Num_ip	
Descripción:	Tabla que contiene el indicador de las diferentes ip.
	• ip • ip_aux

Clave primaria:	• fecha • hora

Atributo	Descripción
ip	Ip versión 4 origen del indicador.
ip_aux	Campo que siempre contiene el valor 'DISTINTAS' para facilitar la visualización y tratamiento realizado posteriormente.
num_veces	Número total de IP destino distintas.
fecha	Fecha en formato YYYY-MM-DD al que corresponde el indicador.
hora	Hora en formato HH al que corresponde el indicador.

Tabla 15 Número total de ip destinos.

Num_app	
Descripción:	Tabla que contiene el indicador de las diferentes ip.
Clave primaria:	• ip • aplicacion • fecha • hora

Atributo	Descripción
ip	Ip versión 4 origen del indicador.
aplicación	Campo que siempre contiene el valor 'APLICACIONES' para facilitar la visualización y tratamiento realizado posteriormente.
num_veces	Número de distintas aplicaciones identificadas.
fecha	Fecha en formato YYYY-MM-DD al que corresponde el indicador.
hora	Hora en formato HH al que corresponde el indicador.

Tabla 16 Número total de las diferentes aplicaciones.

4.5.1.2 Tabla principal.

A continuación se detalla el diseño de la tabla más importante. La importancia de esta tabla es debido a que contiene todas las estadísticas de los cálculos realizados a partir de los datos de las tablas auxiliares.

Estadísticos	
Descripción:	Contiene todos los indicadores así como los datos estadísticos para cada uno de ellos.
Clave primaria:	• ip • indicador • fecha • hora
Atributo	**Descripción**
ip	Ip versión 4 origen del indicador.
indicador	Identidicador del indicador. Para poder comprender la información estos son los posibes valores que se pueden encontrar: • Nombre de una aplicación. • O 'número de puerto' (Indicador de puerto origen). • D 'número de puerto' (Indicador de puerto destino). • I 'IP' (Indicador de icmp). • DISTINTAS (Indicador de distintas ip). • APLICACIONES (Indicador de distintas aplicaciones). • IP (Indicador de relación-ip).
media	Número que corresponde con el cálculo de la media del indicador.
varianza	Número que corresponde con el cálculo de la varianza del indicador.
media_exp	Número que corresponde con el cálculo de la media móvil exponencial del indicador.
varianza_exp	Número que corresponde con el cálculo de la varianza exponencial móvil del indicador.
muestas	Número de muestras del indicador.
fecha	Fecha en formato YYYY-MM-DD al que corresponde el indicador.
hora	Hora en formato HH al que corresponde el indicador.

Tabla 17 Tabla principal de indicadores.

4.5.1.3 Tabla de alarmas.

Si algún valor registrado en las tablas auxiliares no se encuentra dentro de los umbrales, se generará una alarma en la tabla logs registrando tal evento.

Logs	
Descripción:	Contiene todos los eventos generados cuando un indicador ha sido clasificado como anómalo.
Clave primaria:	• ip • indicador • tipo • fecha • hora

Atributo	Descripción
ip	Ip versión 4 origen del indicador.
indicador	Nombre del indicador que ha sido registrado como anómalo.
tipo	Los datos son evaluados a través de dos formas. Por lo que se pueden generar dos tipos de alarmas: • ERROR: Evento que identifica la media. • ERROR_EXP: Evento que identifica la media móvil.
valor	Valor que tiene el indicador cuando se ha clasificado como anómalo.
fecha	Fecha en formato YYYY-MM-DD al que corresponde el indicador.
hora	Hora en formato HH al que corresponde el indicador.

Tabla 18 Logs.

5 PRUEBAS Y RESULTADOS

My work on free software is motivated by an idealistic goal: spreading freedom and cooperation. I want to encourage free software to spread, replacing proprietary software that forbids cooperation, and thus make our society better.

- Richard Stallman -

En este capítulo se verá el diseño del sistema desarrollado, las partes en las que se compone, así como las principales funcionalidades que posee. Cabe destacar que el sistema podría ser usado en casos reales debido a la implementación realizada.

5.1 Apartado

Ksnd´lksajgdlkagl

REFERENCIAS

[1] incibe (Instituto Nacional de Ciberseguridad), "Amenaza vs Vulnerabilidad, ¿sabes en qué se diferencian? | INCIBE," 2017. [Online]. Available: https://www.incibe.es/protege-tu-empresa/blog/amenaza-vs-vulnerabilidad-sabes-se-diferencian. [Accessed: 29-Oct-2018].

[2] P. K. Pateriya and S. S. Kumar, "Analysis on Man in the Middle Attack on SSL," *Int. J. Comput. …*, vol. 45, no. 23, pp. 43–46, 2012.

[3] R. G. Wiliam, "Trujillo_2016," p. 177, 2016.

[4] INCIBE, "Ransomware: una guía de aproximación para el empresario," p. 25, 2017.

[5] M. Barrionuevo, M. Lopresti, N. Miranda, and F. Píccoli, "red usando imágenes y técnicas de Computación de Alto Desempeño .," pp. 1166–1175.

[6] N. Stivet, T. Álvarez, and L. F. Pedraza, "Redes neuronales y predicción de tráfico Neural networks and prediction of traf fic," *Edición Espec.*, vol. 15, no. 29, pp. 90–97, 2011.

[7] Anonymous, "Detecting Anomalies in Communication Packet Streams Based on Generative Adversarial Networks," *Neuropsychology*, vol. 58, no. 6, pp. 1151–1161, 2006.

[8] P. Larrañaga, I. Inza, and A. Moujahid, "Tema 8. Redes Neuronales," *Researchgate*, p. 19, 2015.

[9] R&S, "R & S ® PACE 2 Solution Guide Contents."

[10] T. Firewalls, "Firewalls industriales DPI ¿ Qué es un firewall industrial DPI ? Firewalls industriales DPI," pp. 1–5.

[11] J. J. Dougherty, "Interested in learning more ? In sti tu Au th re ns f rig," *Style (DeKalb, IL)*, no. Security 401, 2011.

[12] E. De La Hoz, E. M. De La Hoz, A. Ortiz, and J. Ortega, "Modelo de detección de intrusiones en sistemas de red, realizando selección de características con FDR y entrenamiento y clasificación con SOM," *Inge Cuc*, vol. 8, no. 1, pp. 85–116, 2012.

[13] E. Arias, "Instituto Politécnico Nacional," *Cic.Ipn.Mx*, pp. 1–80, 2010.

[14] L. R. M, "Snort como herramienta administrativa," no. 5, pp. 74–78.

[15] cvedetails, "CVE security vulnerability database. Security vulnerabilities, exploits, references and more," 2018. [Online]. Available: https://www.cvedetails.com/. [Accessed: 01-Nov-2018].

[16] "Vulnerability distribution of cve security vulnerabilities by types." [Online]. Available: https://www.cvedetails.com/vulnerabilities-by-types.php. [Accessed: 01-Nov-2018].

[17] L. Deri, M. Martinelli, and A. Cardigliano, "nDPI: Open-Source High-Speed Deep Packet Inspection."

[18] sourceforge, "L7-filter Supported Protocols." [Online]. Available: http://l7-filter.sourceforge.net/protocols. [Accessed: 01-Nov-2018].

[19] P. Bulletin, "NBAR2 Protocol Library," pp. 1–47, 2013.

[20] R. Hofstede *et al.*, "Flow Monitoring Explained: From Packet Capture to Data Analysis With NetFlow and IPFIX," *IEEE Commun. Surv. Tutorials*, vol. 16, no. 4, pp. 2037–2064, 2014.

[21] M. De, "Monitorización de una red académica mediante Netflow 1."

[22] R. Antonello *et al.*, "Deep packet inspection tools and techniques in commodity platforms: Challenges and trends," *J. Netw. Comput. Appl.*, vol. 35, no. 6, pp. 1863–1878, 2012.

[23] "Túnel (informática) - Wikipedia, la enciclopedia libre." [Online]. Available: https://es.wikipedia.org/wiki/Túnel_(informática). [Accessed: 04-Nov-2018].

[24] concepto.de, "¿Qué es ISP?" [Online]. Available: https://concepto.de/isp/. [Accessed: 05-Nov-2018].

[25] E. Lgplv, D. Packet, and I. Library, “nDPI - Quick Start Guide,” no. October, pp. 1–15, 2016.

[26] P. Kocher, “Internet Engineering Task Force (IETF) A. Freier Request for Comments: 6101 P. Karlton Category: Historic Netscape Communications,” 2011.

[27] A. V. Aho and M. J. Corasick, “Efficient string matching: an aid to bibliographic search,” *Commun. ACM*, vol. 18, no. 6, pp. 333–340, Jun. 1975.

[28] L. Deri SharkFest, “#sf17eu • Estoril, Portugal How to rule the world… by looking at packets! Turning Wireshark into a Traffic Monitoring Tool: Moving from packet details to the big picture.”

[29] ntop.org, “nProbe documentation — nProbe 8.5 documentation.” [Online]. Available: https://www.ntop.org/guides/nProbe/. [Accessed: 07-Nov-2018].

[30] http://www.tcpdump.org, “Manpage of TCPDUMP.” [Online]. Available: http://www.tcpdump.org/manpages/tcpdump.1.html. [Accessed: 07-Nov-2018].

[31] “Secure Copy - Wikipedia, la enciclopedia libre.” [Online]. Available: https://es.wikipedia.org/wiki/Secure_Copy. [Accessed: 07-Nov-2018].

[32] web.mit.edu, “Protocolo SSH.” [Online]. Available: https://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-ssh.html. [Accessed: 07-Nov-2018].

[33] wikipedia, “Valores separados por comas - Wikipedia, la enciclopedia libre.” [Online]. Available: https://es.wikipedia.org/wiki/Valores_separados_por_comas. [Accessed: 07-Nov-2018].

[34] Elvis Pranskevichus, “What’s New In Python 3.7 — Python 3.7.1 documentation.” [Online]. Available: https://docs.python.org/3/whatsnew/3.7.html. [Accessed: 08-Nov-2018].

[35] “pandas: powerful Python data analysis toolkit Release 0.23.4 Wes McKinney & PyData Development Team,” 2018.

[36] “Guide to Numpy : Travis E. Oliphant : Free Download, Borrow, and Streaming : Internet Archive.” [Online]. Available: https://archive.org/details/NumPyBook/page/n25. [Accessed: 08-Nov-2018].

[37] “8.1. datetime — Basic date and time types — Python 2.7.15 documentation.” [Online]. Available: https://docs.python.org/2/library/datetime.html. [Accessed: 08-Nov-2018].

[38] “os — Miscellaneous operating system interfaces — Python 3.7.1 documentation.” [Online]. Available: https://docs.python.org/3/library/os.html. [Accessed: 08-Nov-2018].

[39] “28.1. sys — System-specific parameters and functions — Python 2.7.15 documentation.” [Online]. Available: https://docs.python.org/2/library/sys.html. [Accessed: 08-Nov-2018].

[40] “MySQL Connector/Python Developer Guide.”

[41] “math — Mathematical functions — Python 3.7.1 documentation.” [Online]. Available: https://docs.python.org/3/library/math.html. [Accessed: 08-Nov-2018].

[42] “10.10. shutil — High-level file operations — Python 2.7.15 documentation.” [Online]. Available: https://docs.python.org/2/library/shutil.html. [Accessed: 08-Nov-2018].

[43] “configparser — Configuration file parser — Python 3.7.1 documentation.” [Online]. Available: https://docs.python.org/3/library/configparser.html. [Accessed: 08-Nov-2018].

[44] “Wijnand Modderman-Lenstra,” 2017.

[45] M. U. Y. Valiosa, “CURSO DE TCP / IP : ICMP (Protocolo de Mensajes de Control de Internet).”

[46] E. W. Weisstein and E. W. Weisstein, “Media aritmética,” *MathWorld*.

[47] wikipedia, “Varianza - Wikipedia, la enciclopedia libre.” [Online]. Available: https://es.wikipedia.org/wiki/Varianza. [Accessed: 15-Nov-2018].

[48] R. G. Brown, *Exponential Smoothing for Predicting Demand*. Cambridge, Massachusetts: Arthur D. Little Inc., 1956.

[49] “MySQL 8.0 Reference Manual - Including MySQL NDB Cluster 8.0.”