Análisis, diseño y simulación de una infraestructura de red segura

Adrian Tamayo Martínez

Resumen– Internet está cada vez más en nuestro día a día trayendo consigo ventajas y desventajas debido a su fácil acceso. Los cibercriminales cada vez tienen un mayor repertorio de métodos y herramientas a su disposición y es por ello que la seguridad es un tema principal a ser tratado en la mayoría de las empresas. Este proyecto consiste en diseñar e implementar una infraestructura de red medianamente segura que permita garantizar la integridad, confidencialidad y disponibilidad de los datos en la red, así como monitorizar y analizar el tráfico de la red en busca de comportamientos anómalos y detección de intrusos.

Palabras clave– Internet, Seguridad, Firewall, Iptables, VPN, DMZ, HAProxy, Metasploit, Snort, Nessus, NMap, Nagios.

Abstract– Internet is increasingly in our day to day bringing with it avantages and disadvantages due to its easy accessibility. Cybercriminals increasingly have a greater repertoire of methods and tools at their disposal and that is why security is a main issue to be addressed in most companies. This project consists of designing and implementing a moderately secure network infrastructure that guarantees the integrity, confidentiality and availability of network data, as well as monitoring and analyzing network traffic in search of anomalous behavior and intrusion detection.

Keywords– Internet, Security, Firewall, Iptables, VPN, DMZ, HAProxy, Metasploit, Snort, Nessus, NMap, Nagios.

1 Introducción

La aparición de Internet ha supuesto un antes y un después en la comunicación entre las personas, trayendo consigo ventajas y desventajas. Si retrocedemos unos años atrás, las personas se comunicaban por medios como cartas, llamadas telefónicas, en persona, con el objetivo de compartir información, no obstante, este tipo de medios eran lentos. Con la aparición de Internet, ha supuesto una evolutiva mejora a la hora de comunicarse, y con ello, proporcionar una gran cantidad de datos e información al alcance de cualquiera.

Con toda la información que circula por Internet, surge la necesidad de proteger dicha información y los sistemas que la contienen. Esta información desprotegida es el objetivo de muchos cibercriminales y organizaciones, y para ello es importante establecer medidas de seguridad para protegerla. Por eso, un administrador de red debe de garantizar la integridad, confidencialidad y disponibilidad de la información en su sistema.

El proyecto consiste en analizar, diseñar y simular una infraestructura de red segura formada por tres sedes empresariales diferentes. El proyecto se ha estructurado de la siguiente forma: en primer lugar, se detallan los objetivos principales a ser desarrollados para la realización del proyecto. Seguidamente, se explica el estado del arte del proyecto, analizando las herramientas más usadas y efectivas para implementar seguridad en el sistema. A continuación, se describe la estructuración del proyecto. Después, se detalla la metodología utilizada y llevada a cabo para la realización del proyecto. En el primer módulo, se muestra el diseño definitivo de la infraestructura de red y los programas empleados para ello, así como, el uso de la herramienta IPtables para la configuración de seguridad en los firewalls de la red y de la creación de un túnel de comunicación entre las diferentes sedes, mediante la implementación de IPSec. En el segundo módulo, se explica la monitorización y análisis del estado de la red mediante el uso de herramientas como Nagios y HAProxy, y la detección de intrusos con Snort. En el tercer módulo, se realizan los análisis y explotación de las vulnerabilidades más comunes en los sistemas de red. Seguidamente, se muestran los resultados obtenidos en las

• E-mail de contacte: Adrian.TamayoM@e-campus.uab.cat
• Menció realitzada: Tecnologies de la Informació
• Treball tutoritzat per: Juan Carlos Sebastián Pérez (dEIC)
• Curs 2018/19

secciones anteriores. A continuación, se expone de forma conjunta las conclusiones extraídas sobre la realización del proyecto y de todas las herramientas utilizadas. Después, se explica las posibles líneas de futuro que puede tener dicho proyecto. Por último, se presentan los agradecimientos y las referencias consultadas para la elaboración del proyecto.

2 OBJETIVOS

El objetivo principal de este proyecto es analizar, diseñar y simular una infraestructura de red enfocada al mundo empresarial teniendo en cuenta las demandas de seguridad que existen actualmente. Dicha infraestructura de red estará compuesta por 3 sedes empresariales. Para poder llevar a cabo el proyecto, los principales objetivos a cumplir son los siguientes:

- Diseño del diagrama de red de la empresa.
- Monitorización del estado normal de funcionamiento de la red.
- Creación de un Firewall y/o una Zona Desmilitarizada (DMZ).
- Análisis de rendimiento.
- Implementación de VLANs.
- Análisis de vulnerabilidades.
- Balanceo de carga.
- Backups en red.
- Monitorización y detección de alguna intrusión mediante el uso de herramientas como Snort o Nagios.

3 ESTADO DEL ARTE

Actualmente, los administradores de redes, gracias a la existencia de Internet, tienen a su disposición un gran despliegue de herramientas gratuitas y comerciales, útiles para su correcta y eficiente administración.
Un administrador debe de plasmar el diseño de red pensado en una herramienta de diseño, como puede ser Lucidchart (utilizada en este proyecto), Visual Paradigm, Pencil o Draw.io. Para simular el funcionamiento de la red, se tiene al alcance herramientas como CORE, disponible para sistemas operativos basados en Linux. En el proyecto se ha utilizado la versión 4.7. Otras herramientas de simulación que ofrecen un funcionamiento parecido a CORE son Cisco Packet Tracert, GNS3, Packet Tracert, entre otras.
Para la monitorización del tráfico de la red tenemos a Nagios, herramienta más utilizada por los administradores de red y la herramienta Snort, unos de los detectores de intrusos más utilizados. En el proyecto se han utilizado la versión Nagios XI y la versión 2.9.13 de Snort. En la configuración e implementación de túneles y VPN existen una gran variedad de formas, entre ellas el uso del conjunto de protocolos IPSec. Para la configuración de los firewalls, la herramienta Iptables (versión 1.4.14), un cortafuegos instalado en los sistemas operativos Linux, que, con una sintaxis muy básica, permite gestionar el tráfico de entrada y salida del equipo y/o red.
Para la detección de vulnerabilidades del sistema, tenemos a disposición la herramienta Nessus juntamente con el uso de la herramienta Nmap, para uso en terminal o Zenmap para uso en interfaz gráfica. En una infraestructura de red es importante balancear la carga de peticiones que se producen en los servidores, para ello la herramienta idónea es el balanceador de carga HAProxy, en su versión 2.0.

4 METODOLOGÍA

La metodología escogida para llevar a cabo el desarrollo del proyecto ha sido una metodología ágil basada en Sprints de Scrum [1]. Esta metodología es iterativa e incremental. Cada iteración o sprint suele tener, de media, una duración de entre 2 a 4 semanas, pero no debe sobrepasar el mes de duración. En este caso, la duración de cada sprint estaba asociado a cada una de las diferentes entregas parciales del proyecto. Un equipo de Scrum está formado por los siguientes roles:

- **Product Owner**: es el rol central del proyecto. Responsable de los requerimientos, decide el Product Backlog y cambia y prioriza este antes de cada sprint.
- **Scrum Master**: persona encargada de liderar al equipo y de realizar el sprint de forma correcta coordinando los diferentes miembros, eliminando obstáculos del equipo. Hace de hilo comunicativo entre el Product Owner y el Scrum Team.
- **Scrum Team**: grupo de profesionales con los conocimientos técnicos necesarios para desarrollar el proyecto.

Como el proyecto a desarrollar es individual, y con la supervisión de un tutor académico, los roles se han distribuido de la siguiente forma: el tutor hará la función de Product Owner (Cliente) y el alumno de Scrum Master y Scrum Team, ya que será el alumno el encargado de realizar el documento de requisitos a partir de las especificaciones del tutor, de planificar los sprints, realizar las diferentes tareas y de supervisar la realización de las tareas. Scrum tiene diferenciadas las diferentes etapas en la que se compone:

- **Product Backlog**: lista de requerimientos/objetivos del proyecto. Pertenece al Product Owner.
- **Sprint**: subconjunto de tareas extraídas del Product Backlog. Un sprint no debe de sobrepasar las 300 tareas y es creado solamente para el Scrum Team.
- **Sprint Panning**: proceso en el que se seleccionan las tareas a ser realizadas en el sprint. No debe sobrepasar las 8 horas de duración.
- **Sprint Review**: reunión que se realiza al final de cada sprint, en este caso, en cada entrega de informe. En la reunión se analiza el trabajo realizado con el tutor y se planifican, descartan o se incluyen nuevas tareas para el siguiente sprint.

Para poner en práctica la metodología Scrum se ha utilizado la herramienta Azure DevOps.

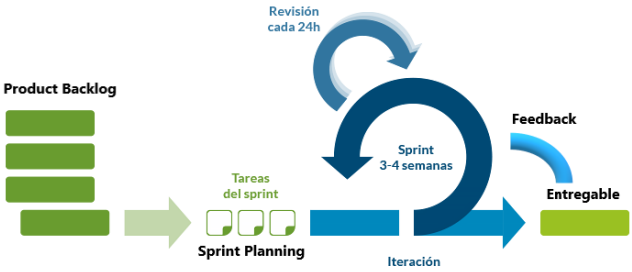

Fig. 1: Etapas de la metodología Scrum.

5 Diseño y configuración de la infraestructura de red

En esta sección se explica el diseño de la infraestructura de la red, así como los programas utilizados para ello y la implementación de la configuración de la misma.

5.1. Diseño

El esbozo principal de la infraestructura de red, como se puede observar en la Fig. 2, se ha realizado mediante el uso de una herramienta gratuita en línea llamada Lucidchart [2], la cual proporciona una gran variedad de tipos de diagramas y elementos de red para confeccionarlos. Una vez se tiene

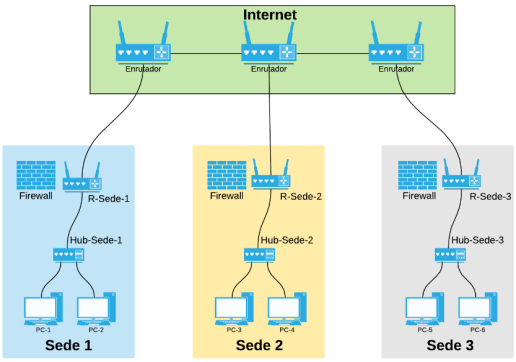

Fig. 2: Diagrama de la infraestructura de red.

el esbozo de la infraestructura de red, es hora de pasarlo a un simulador de redes.

5.2. CORE

Common Open Research Emulator (CORE) es una herramienta de código libre para emular redes en una o más máquinas. Puede conectar estas redes emuladas a redes en vivo. CORE consta de una GUI para dibujar topologías de máquinas virtuales ligeras y módulos de Python para la emulación de red de scripts [3]. CORE ha sido desarrollado por un grupo de investigación de tecnología de redes que forma parte de la división de investigación y tecnología de Boeing.
Actualmente, el proyecto tiene el soporte del Laboratorio de Investigación Naval de los Estados Unidos. Algunas de las mejores características que proporciona esta herramienta son:

- Laboratorio de red eficiente y escalable
- Interfaz de usuario intuitiva y fácil de utilizar
- Configuración y controles centralizados
- Ejecución de protocolos y aplicaciones sin ser modificados
- Conexión a tiempo real a redes en vivo
- Altamente personalizable

En la Fig. 3, se puede observar el diseño final de la infraestructura de red en la herramienta CORE. Dicha herramienta ha sido instalada en una máquina virtual Debian 7.X de 32 bits.

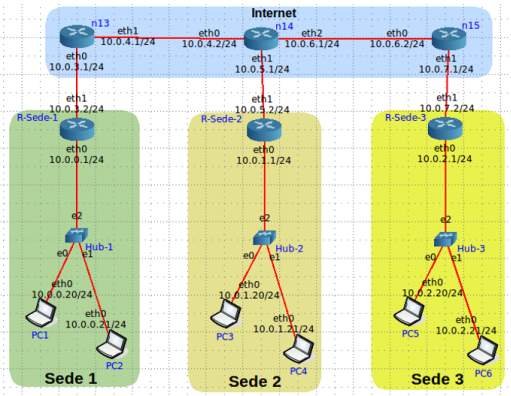

Fig. 3: Diagrama final de la infraestructura de red mediante el uso de la herramienta CORE.

5.3. Iptables

La implementación de los firewalls de la infraestructura de red se ha utilizado la herramienta de Iptables. Iptables es una herramienta avanzada de filtrado de paquetes integrada en el núcleo de Linux, la cual se encarga de analizar y decidir que hacer con cada uno de los paquetes entrantes en una máquina en función de un conjunto de reglas establecidas [4]. Algunas de las acciones que se puede realizar con Iptables son las siguientes:

- Implementar un cortafuegos (firewall)
- Configurar un dispositivo como traductor de direcciones de red (NAT - *Network Address Translation*)
- Marcar y modificar paquetes
- Registrar registros (logs) del tráfico de red
- Depurar el funcionamiento de la red

Como se ha mencionado anteriormente, Iptables permite definir un seguido de reglas sobre qué acción aplicar a cada paquete. Las reglas son agrupadas en cadenas, donde cada una de estas contiene una lista de cadenas. Las cadenas se agrupan en tablas, donde cada tabla está asociada a un tipo diferente de procesamiento de paquetes. Iptables permite crear nuevas tablas, crear y eliminar cadenas, excepto las predefinidas. Esta herramienta incorpora tres tablas predefinidas [5]:

- **Filtrado** (*filter*): tabla por defecto. Tabla más utilizada y es la responsable del filtrado de los paquetes según se ha configurado el firewall. Todos los paquetes pasan a través de esta tabla. Está formada por las cadenas: **INPUT**: para la entrada. Todos los paquetes destinados a entrar en nuestro sistema deben pasar por esta cadena. **OUTPUT**: para la salida. Todo los paquetes creados por el sistema y que van a salir hacia otra máquina. **FORWARD**: redireccionamiento.

- **NAT**: esta tabla será consultada cuando un paquete crea una nueva conexión. Permite compartir una dirección IP pública entre muchas otras máquinas. Con ella se puede añadir reglas para modificar las direcciones IP de los paquetes y contienen dos reglas: SNAT para la dirección de origen y DNAT para las direcciones de destino. Esta tabla contiene las cadenas: **OUTPUT**, **PREROUTING**: para modificar los paquetes tan pronto llegan a la máquina. **POSTROUTING**: para modificar los paquetes que estan listos para salir de la máquina.

- **Mangle**: esta tabla se encarga de modificar las opciones del paquete (*TOS (Type Of Service), TTL (Time To Live) o MARK*). Todos los paquetes pasan por esta tabla. Esta tabla esta compuesta por las cadenas **INPUT**, **OUTPUT**, **FORWARD**, **PREROUTING** y **POSTROUTING**.

5.4. VPN: Tunnelling y IPSec

En ocasiones se puede dar el caso de que entre usuarios de diferentes sedes quieran intercambiar información comprometida de sus empresas. La estructura de la red actual, para comunicarse entre las sedes, la información debe de viajar por Internet siendo esta vulnerable a ser interceptada por terceras personas.
Para solventar este problema existen diferentes técnicas, entre ellas, la implementación de una red virtual privada (*Virtual Private Network - VPN*). Una VPN consiste en crear una conexión directa ficticia entre dos puntos, en este caso, entre las dos sedes, mediante un túnel. Esta técnica puede ser utilizada para evitar o bloquear un firewall. Para ello se encapsula el protocolo bloqueado dentro de otro permitido, habitualmente se utiliza el protocolo HTTP [6].
Con el fin de poder minimizar los riesgos y poder garantizar la seguridad de la información se crearon un conjunto de protocolos de seguridad llamado Protocolo de Seguridad de Internet (del inglés *Internet Protocol Security* – IPSec) [7]. Al combinar los protocolos, proporciona a la transmisión de paquetes de datos una seguridad más fiable. IPSec, como ya se ha comentado anteriormente, es un conjunto de protocolos cuya arquitectura ha sido propuesta como estándar por el Grupo de Trabajo de Ingeniería de Internet (IETF), una organización dedicada al desarrollo técnico de Internet. IPSec está disponible para los usuarios desde la última versión del Protocolo de Internet (IPv6) y fue desarrollado posteriormente para IPv4 dividiéndose en tres grandes grupos:

- **Protocolos de transferencia**:
  - *Authentication Header* - AH
  - *Encapsulating Security Payload* - ESP
- **Gestión de claves**:
  - *Internet Security Association and Key Management Protocol* - ISAKMP
  - *Internet Key Exchange* - IKE
- **Base de datos**:
  - *Security Association Database* – SAD
  - *Security Policy Database* – SPD

Con la ayuda de los protocolos de transferencia AH y ESP, IPSec garantiza la autenticidad e integridad de los datos enviados, asegurando que su contenido no haya sufrido cambios desde que fue enviado hasta su posterior recepción. Para este fin, AH ofrece una extensión de la cabecera concentrándose, por un lado, en la autentificación, y por otro, en evitar que los paquetes sean manipulados durante el proceso de la transmisión, añadiendo un número de secuencia en la cabecera.

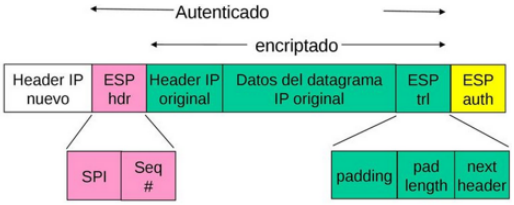

Fig. 4: Ejemplo de un datagrama bajo el uso de IPSec.

Además de comprobar la integridad y la identidad, el protocolo ESP cifra los datos enviados. El protocolo IKE es el principal responsable de la gestión del cifrado en ESP. Para garantizar estas medidas de seguridad entre emisor y receptor, IKE utiliza el método Diffie Hellman para un intercambio seguro de claves. Actualmente existen dos modos de transferencia para establecer conexiones seguras: el modo transporte, en el que los dos puntos finales (endpoint) están conectados directamente, y el modo túnel, en donde se crea una conexión entre dos redes IP. Para la realización de este proyecto se ha implementado un túnel en modo transporte.

5.5. DMZ

Una Zona Desmilitarizada (del inglés *Demilitarized Zone*, DMZ) [8] es una red local que se ubica entre la red interna de una organización y una red externa, generalmente en Internet. El objetivo de una DMZ es que las conexiones desde la red interna y la externa a la DMZ esten permitidas, mientras que desde la DMZ solo este permitido las conexiones hacia la red externa. Esto permite que las DMZ puedan dar servicios a la red externa a la vez que protegen la red interna en caso de que intrusos comprometan la seguridad de los equipos situados en la DMZ.

6 MONITORIZACIÓN, DETECCIÓN DE INTRUSOS Y ANÁLISIS DEL RENDIMIENTO DE LA INFRAESTRUCTURA DE RED

En la infraestructura de red implementada existe un tráfico continuo de datos entre las diferentes redes que crean las sedes e Internet. Los cibercriminales suelen dedicarse

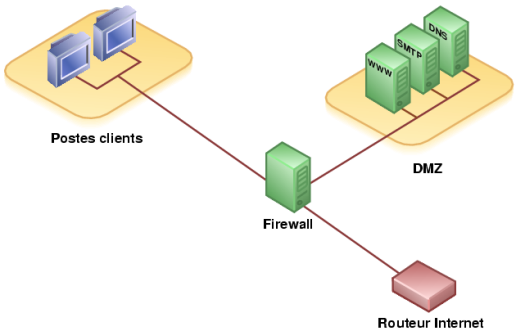

Fig. 5: Ejemplo de ubicación de una red DMZ.

a atacar a dichos servidores y, por ello, hay que implementar herramientas de detección para anticiparse a los posibles ataques que puedan realizarse. Hoy en día tenemos a nuestra disposición un gran despliegue de herramientas que están a nuestro alcance y que podemos utilizar. En esta sección se mostrará el uso de la herramienta Snort para la monitorización del tráfico de la red y la detección de accesos prohibidos a determinadas webs y la detección de comandos como *ping* o escaneo de puertos, entre otros. También se explicará otra herramienta importante de monitoraje y detección de comportamiento anómalo de la red, como es Nagios y se darán algunas pinceladas sobre el uso de HAProxy para el balanceo de carga del tráfico de la red.

6.1. Snort

Snort es un sistema de detección de intrusos (IDS por sus siglas en inglés *Intrusion Detection System*) de código abierto. Un IDS intenta detectar o monitorizar los eventos ocurridos en un determinado sistema informático o red en busca de intentos de comprometer la seguridad de dicho sistema en tiempo real [9]. Una función de los IDS es buscar patrones previamente definidos que impliquen cualquier tipo de actividad sospechosa sobre nuestra red o equipo. Por lo tanto, aportan a nuestra seguridad una capacidad de prevención y de alerta anticipada ante cualquier actividad sospechosa. Cabe decir que los IDS no están diseñados para detener un ataque, aunque, sí pueden generar ciertos tipos de respuesta ante estos [10].

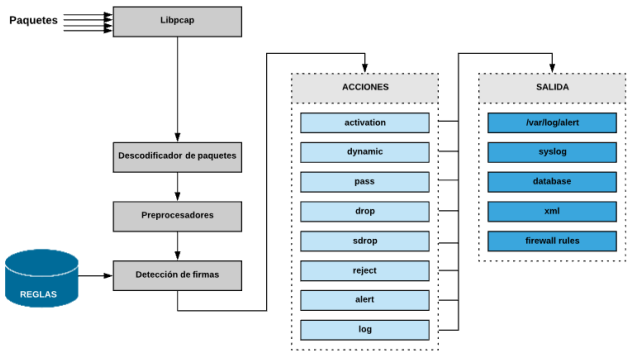

Fig. 6: Diagrama sobre la arquitectura de la herramienta Snort.

Existen diferentes tipos de IDS [11]:

- **Host IDS (HIDS)**: se encarga de proteger un único servidor o equipo. Monitorizan gran cantidad de eventos y recaban información del sistema como ficheros, logs, recursos, etc, para su posterior análisis en busca de posibles incidencias. Todo se realiza de modo local.
- **Net IDS (NIDS)**: se encarga de proteger un sistema basado en red capturando y analizando paquetes red actuando como *sniffers*. Este tipo de IDS no sólo trabaja a nivel TCP/IP, sino que también lo pueden hacer a nivel de aplicación.
- **Híbridos**: por el tipo de respuesta se pueden clasificar en:
  - **Pasivos**: IDS que notifica al administrador de red mediante el uso de alertas, logs, etc. Pero no actúan sobre el ataque.
  - **Activos**: IDS que genera algún tipo de respuesta sobre el ataque como, por ejemplo, cerrar la conexión o enviar algún tipo de respuesta predefinida en nuestra configuración.

La arquitectura de este sistema se basa en un lenguaje de creación de reglas en el que se pueden definir los patrones que se utilizarán a la hora de monitorizar el sistema. Snort ya ofrece una serie de reglas y filtros predefinidos que se pueden ajustar durante su instalación y/o configuración para adaptarlo lo máximo posible a lo que deseamos.
Para la realización del proyecto, se han creado una serie de reglas que permiten detectar accesos a páginas web prohibidas como Youtube.com, *pings* hacia la máquina, escaneos de puertos utilizando Nmap y conexiones ssh. Estas reglas deben de incluirse en el directorio */etc/snort/rules*. Este directorio lo utiliza Snort por defecto, una vez inicializado, para activar todas las reglas configuradas. El archivo de configuración se encuentra en el directorio */etc/snort*. En este archivo, solamente debemos de incluir nuestras reglas, en la sección *'Step #7: Customize your rule set'* implementadas de la siguiente forma: *#include $RULE_PATH/[nombre-archivo].rules*

6.2. Nagios

Nagios es un sistema de monitorización de equipos y servicios de red. Esta implementado en lenguaje C y bajo la licencia *General Public License*, GNU [12]. Vigila los equipos y servicios que se especifican, alertando cuando el comportamiento de los mismos no es el deseado. Permite tener un completo control de la disponibilidad de los servicios, procesos y recursos de las máquinas, informando al administrador de red los problemas incluso antes de que los usuarios se den cuanta, de forma que se pueda actuar de forma pro-activa.
Entre sus características principales cabe destacar:

- Monitorización de servicios de red (SMTP, POP3, HTTP, etc)
- Monitorización de los recursos del sistema (uso de discos, memoria, estado de los puertos, etc)
- Independecnia de sistemas operativos
- Posibilidad de monitorización remota mediante túneles SSL cifrados o SSH

- Posibilidad de implementar plugins específicos para nuevos sistemas

Nagios proporciona una gran versatilidad para consultar prácticamente cualquier parámetro de interés de un sistema, y genera alertas, que pueden ser recibidas mediante correo electrónico, mensajes SMS, entre otros, cuando los parámetros exceden los márgenes definidos por el administrador de red. Para la realización del proyecto, se ha tenido que configurar cada dispositivo de la red en el directorio */etc/nagios3/objects*, por ejemplo, para la configuración del router de la sede 1 se ha creado un archivo de configuración en */etc/nagios3/objects/routerSede1.cfg*. Para este y cada uno de los routers se ha creado un servicio que realiza pings cada 5 minutos y si falla los realiza cada minuto, devolviendo una alerta de tipo crítica cuando el ping tarda más de 600ms y una alerta normal cuando supera los 200ms. Dichas alertas se notifican tanto en la interfaz gráfica como por correo electrónico.

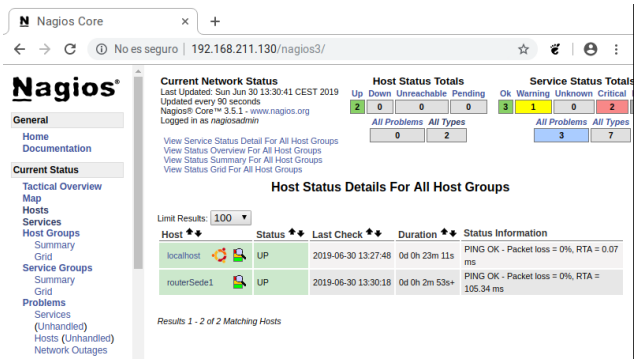

Fig. 7: Estados de los diferentes equipos de la red, en este caso del routerSede1 y un pc de dicha sede.

6.3. Balanceo de carga - HAProxy

En algunas ocasiones, el volumen de tráfico de la red puede sobrepasar el umbral establecido por el administrador de red por una mala implementación o por una mala estimación, ocasionado una sobrecarga de los servidores. La función principal de un balanceador de carga es distribuir el tráfico de la red entre varios servidores, asegurando así, la disponibilidad de los datos en la red.
Los balanceadores de carga se agrupan en dos categorías [13]:

- **Layer4**: actúan sobre los datos de la red y protocolos IP, TCP, FTP y UDP.
- **Layer7:**: distribuyen peticiones en la capa de aplicación con protocolos como HTTP o TCP.

Actualmente existen un gran nombre de balanceadores de carga, pero uno de los más utilizados es HAProxy, del inglés *High Availability Proxy*. HAProxy es un software de código libre que actúa como balanceador de carga ofreciendo alta disponibilidad, balanceo de carga y proxy para comunicaciones TCP y HTTP. Este software está pensado especialmente para balanceadores de tipo Layer7. Las peticiones recibidas por el balanceador son distribuidas entre los servidores disponibles mediante el uso del algoritmo Round Robin. Este algoritmo, está definido como el algoritmo por defecto a utilizar, va seleccionando por turnos el servidor que atenderá la petición, de manera que las peticiones se distribuyen uniformemente entre todos los servidores disponibles de la red.
El archivo de configuración de HAProxy se encuentra en la ruta */etc/haproxy*. En este archivo se han incluido los servidores web en la sección del *Backend* e indicamos el puerto donde se da servicio a la página de estadísticas, en este caso, se ha indicado el puerto 8800 con la directriz *bind *:8800*. Para añadir seguridad al acceder a dicha web, se ha añadido la directriz *stats auth adrian:adrian*, la cual al ingresar en la dirección web *http://[IP-haproxy]:8800* se pedirá unas credenciales de acceso, en este caso, adrian:adrian.

7 Análisis y explotación de vulnerabilidades del sistema

En esta sección se va a tratar de dar a conocer las vulnerabilidades más comunes que pueden encontrarse en una infraestructura de red o sistema y las explotaciones de las mismas. Para el análisis y explotación de las vulnerabilidades se ha utilizado el framework de Metasploit y una máquina virtual basada en Linux diseñada intencionadamente para analizar y explotar vulnerabilidades comunes, Metasploitable 2.

7.1. Vulnerabilidades del sistema

7.1.1. Escaneo de puertos

Una de las primeras actividades que un cibercriminal realizará contra su máquina objetivo será un escaneo de puertos, ya que, de esta forma, podrá obtener información básica acerca de qué servicios se están ofreciendo en la máquina objetivo y otro tipo de detalles como el sistema operativo instalado en la máquina o ciertas características de la arquitectura de la red.
Comprobar el estado de un determinado puerto es a priori una tarea muy sencilla, incluso es posible llevarla a cabo desde un terminal. Existen herramientas como Nmap que pueden realizar dicha tarea de una forma más eficiente y automatizada. Es por ello, que implementar en el firewall medidas contra a estas actividades es fundamental para dificultar al atacante la tarea del escaneo. Un ejemplo de regla de Iptables que sirve como medida de seguridad es: *iptables -A INPUT -m recent –name portscan –rcheck –seconds 86400 -j DROP*. Esta regla, al detectar un escaneo de puertos mediante el flag ”*portscan*”bloquea la dirección IP del atacante un tiempo asignado, en este caso 86400 segundos (24 horas).

7.1.2. Ataques de diccionario

Los ataques de diccionario consisten en recuperar las credenciales de acceso de la máquina objetivo mediante consecutivos intentos de combinaciones posibles de usuario:contraseña, hasta dar con las credenciales correctas. En este tipo de ataques suelen existir tres tipos de ficheros: en primer lugar, un fichero con los nombres de usuarios más comunes, utilizados y los propios que incluye el atacante. En segundo lugar, un fichero con las contraseñas más usadas, más comunes y las propias añadidas por el atacante.

Por último, un fichero con todas las combinaciones posibles de usuario:contraseña de los dos primeros ficheros. Este tipo de ataque no se realiza de forma rápida, ya que, por una parte, depende de la complejidad añadida en el usuario y/o contraseña, y por otro, de la capacidad de operación de la máquina atacante.
Una medida de seguridad a este tipo de ataques es implementar nuevas reglas en el firewall, asignando un número máximo de intentos de conexión, bloqueando la dirección IP atacante si se supera dicho número máximo establecido.

7.1.3. Ataques de Buffer Overflow

Una de las vulnerabilidades más populares y quizás una de las más explotadas han sido los ataques de desbordamiento de búfer, del inglés *Buffer Overflow*.
Un Buffer Overflow se define como la condición en el cual un programa intenta escribir datos más allá de los límites impuestos por un buffer de mida fija. Dada esta vulnerabilidad, podemos llegar a alterar el flujo de ejecución de un programa o incluso ejecutar trozos arbitrarios de código [14]. El resultado de modificar la dirección de retorno es obtener una terminal con privilegios de administrador. Esto conlleva, a que un desbordamiento de los datos del programa puede causar una modificación del flujo de ejecución alterando la dirección de retorno del mismo programa.
En los últimos 3 años, en 2017 se registraron un total de 2797 vulnerabilidades, en 2018 un total de 2487 y en actualmente en lo que lleva de 2019 hay registradas 619 vulnerabilidades [15]. Con los datos expuestos anteriormente y con los datos que se proporcionan en la Fig. 8, este tipo de ataque está considerado como el 3r tipo de ataque más utilizado por los atacantes.

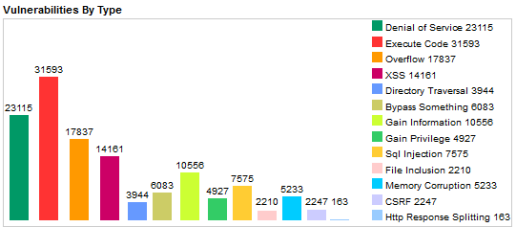

Fig. 8: Gráfico que muestra la cantidad de vulnerabilidades registradas por tipo de ataque.

7.2. Metasploit Framework

Metasploit es un proyecto de código abierto desarrollado para la investigación de vulnerabilidades de seguridad en los equipos de un sistema. Metasploit Framework es una herramienta desarrollada, en su mayor parte, con los lenguajes de programación Perl y Ruby [16]. Esta herramienta está enfocada a las auditorias de seguridad y a equipos Red Team y Blue Team. El Red Team es el equipo ofensivo y encargado del hacking ético haciendo pruebas de intrusión, mientras que el Blue Team es el equipo que lleva a cabo la seguridad y toda la parte defensiva del sistema.
Metasploit framework es una herramienta muy completa que tiene una gran colección de exploits, que son vulnerabilidades conocidas, en las cuales tienen también unos módulos llamados payloads, que son los códigos que explotan estas vulnerabilidades. Una de sus muchas características y ventajas de este framework, es que permite la interacción con otras herramientas externas como Nmap o Nessus.
Cabe destacar que es multiplataforma y gratuita, aunque existe una versión comercial, Metasploit Pro. Para el desarrollo de este proyecto se ha utilizado la última versión gratuita disponible para equipos Linux, Metasploit 4.15.0.

7.3. Nessus

Nessus es una potente herramienta de análisis y detección de posibles vulnerabilidades, fácil de utilizar, basándose en una amplia base de datos de plugins, la cual se actualiza diariamente.
Básicamente Nessus se utiliza como asistente de mantenimiento, es decir, se utiliza para comprobar la seguridad y encontrar vulnerabilidades, para que puedan ser solucionadas por el administrador del sistema [17]. Consiste en un demonio, llamado *nessusd*, encargado de realizar el escaneo en la máquina o sistema objetivo, y Nessus, el cliente, el cual está basado en consola o mediante una interfaz gráfica, que muestra el avance e información sobre el estado de los escaneos.
Nessus tiene un sistema de clasificación basado en el grado de vulnerabilidad: **Crítica**, **Alta**, **Media**, **Baja** y **Información**. Algunas de las características de Nessus son las siguientes:

- Permite conocer qué brechas de seguridad pueden tener los servicios susceptibles a ataques.
- Funciona mediante un proceso de alta velocidad por el que encuentra los datos sensibles y trabaja con la auditoria de configuraciones.
- El servidor realiza todo el trabajo de escaneo que especifica el administrador.
- Indica la vulnerabilidad existente e indica como explotar esta y como proteger al equipo de ella.

En la Fig. 9, se puede observar la cantidad de vulnerabilidades encontradas en la máquina objetivo. Como ya se ha mencionado anteriormente Nessus clasifica las vulnerabilidades según su grado de vulnerabilidad: Crítica: rojo, Alta: naranja, Medio: amarillo, Bajo: Verde y Información: azul.

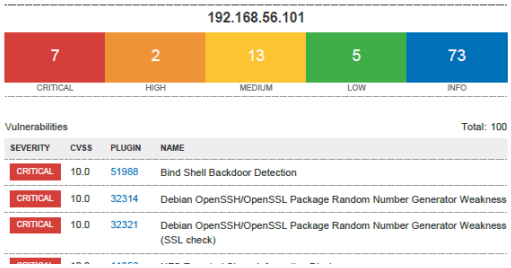

Fig. 9: Vulnerabilidades encontradas y clasificadas en la máquina objetivo utilizando Nessus.

7.4. CVE

Common Vulnerabilities and Exposures (CVE) es una lista de identificadores comunes y únicos para vulnerabilida-

des de ciberseguridad conocidas públicamente [18]. Las entradas CVE son asignadas por las Autoridades de Numeración CVE (CNA) de todo el mundo. Estas entradas proporcionan una base de referencia para la evaluación de herramientas y permite automatizar datos. La lista fue definida y mantenida por *The MITRE Corporation*.
Cada entrada de CVE incluye [19]:

- **Número de identificación**: CVE-YYYY-NNNN, donde YYYY indica el año y NNNN el número de vulnerabilidad, un ejemplo de número de identificación seria el siguiente: CVE-2019-001606.
- **Descripción**: se incluye una breve descripción de la vulnerabilidad o exposición de la seguridad.
- **Referencia**: se incluye cualquier referencia pertinente, es decir, informes de vulnerabilidad, avisos, etc.

El proceso de creación de una entrada CVE es el siguiente: se comienza con el descubrimiento de una vulnerabilidad de seguridad potencial. A la información recogida se le asigna un número de identificación por parte de una CNA. La CNA escribe una descripción y añade las referencias. Por último el equipo de CVE agrega la entrada CVE completa a la lista de CVE y la publica en la página web de CVE.
A cada vulnerabilidad identificada, a parte de asignarle un número de identificación, también se le asigna el nivel de impacto que esta puede ocasionar sobre el sistema o máquina mediante un sistema de puntuación, *Common Vulnerability Scoring System*, CVSS. La puntuación asignada a la vulnerabilidad va en función del nivel de peligrosidad.

8 RESULTADOS

En esta sección se mostrarán los resultados prácticos de las técnicas y tipos de ataques que se han mencionado en las secciones anteriores. Los resultados se mostrarán conforme los tres módulos que conforman el proyecto.

8.1. Diseño y configuración de la infraestructura de red

8.1.1. VPN: Tunneling y IPSec

Para la realización de la VPN entre las sedes 1 y 2 se ha creado otra infraestructura de red partiendo de la infraestructura de red original para simular un escenario solamente con las dos sedes.
El túnel implementado entre los Routers RSEDE1 y RSEDE2 utiliza IPSec, con lo cual todo los paquetes que utilicen el túnel estar cifrados y autentificados. En la Fig. 10 podemos observar un paquete interceptado en RSEDE2 tras realizar un ping de PC1 a PC3 mediante Whireshark. En el paquete se puede observar que se utilizan las cabeceras de IPSec, ESP (cifrado) y AH (autentificación).

8.2. Monitorización, detección de intrusos y análisis del rendimiento de la infraestructura de red

8.2.1. Snort

Para un administrador de red es muy importante monitorizar todo el tráfico entrante y saliente de la red para detectar

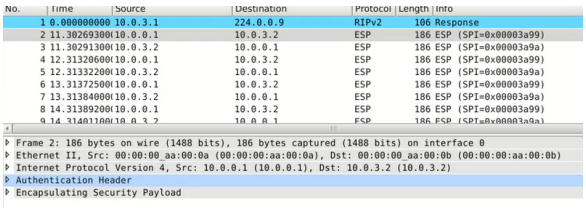

Fig. 10: Paquete cifrado interceptado en Internet entre PC1 y PC3.

cualquier tipo de actividad sospechosa y actuar al respecto. Muchas empresas, tienen como norma no dejar que los trabajadores visiten páginas ajenas a las del trabajo como puede ser facebook, youtube, etc. Snort es una herramienta idónea para detectar dicho comportamiento en la red y alertar al administrador de red si hay algún trabajador visitando páginas webs prohibidas.
En la Fig. 11 podemos observar la alerta al acceder a una página web prohibida, así como, alertas de seguridad de detección de pings y conexiones ssh. En dichas alertas se indica la dirección IP de la máquina atacante.

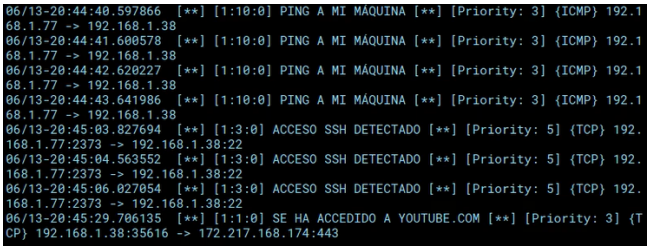

Fig. 11: Alertas recibidas por el administrador de red mediante el uso de Snort.

8.2.2. HAProxy

Disponer de respuesta a una petición de un servicio solicitado es una tarea que todo administrador de red debe de poder garantizar. HAProxy es un balanceador de carga que permite realizar esta configuración, ya que distribuye las peticiones uniformemente entre todos los servidores disponibles de la red, proporcionando así un balanceo de carga equilibrado y estable. Para realizar el balanceo de carga, se ha instalado HAProxy en el Router de cada sede y Apache2 en cada uno de los servidores. Para mostrar el uso de HAProxy se ha utilizado el router y dos servidores (webs1 y webs2) de la sede 1. En la Fig. 12 se puede observar una captura sobre el estado en que se encuentran los servidores, así como el número de peticiones que se han realizado a cada servidor, número de bytes de entrada y salida, etc. Para comprobar que el balanceo de carga entre los dos servidores funciona correctamente, en una pestaña nueva del navegador, se ha accedido a la dirección IP correspondiente al servicio de HAProxy, en este caso, *http://192.168.1.38*. Como se puede observar en la Fig. 13, al acceder a dicha web se muestra la web que proporciona el primer servidor (a). Si accedemos de nuevo a la misma dirección web o si recargamos de nuevo la página, en este caso, se mostrará la web que proporciona el segundo servidor (b), mostrando que el balanceo de carga entre los dos servidores funciona correctamente. Para diferenciar la misma página web en

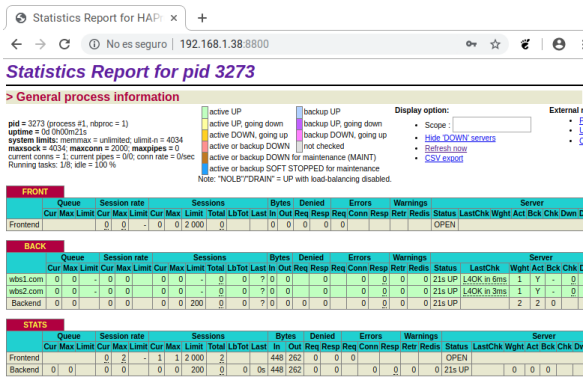

Fig. 12: Informe de estadísticas de HAProxy.

los dos servidores, se ha añadido un título indicando en que servidor se está haciendo la petición.

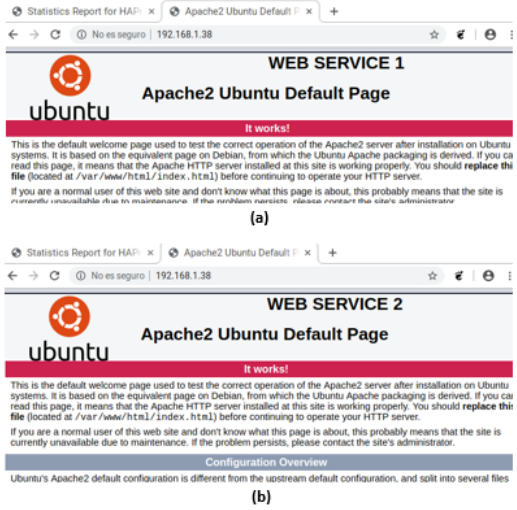

Fig. 13: Páginas web de los diferentes servidores al realizar balanceo de carga con HAProxy.

8.3. Análisis y explotación de vulnerabilidades del sistema

8.3.1. Escaneo de puertos

El escaneo de puertos es una de las primeras actividades que realizan los cibercriminales para obtener información de la máquina objetivo. Mediante el uso de la interfaz gráfica de Nmap, Zenmap, se ha realizado un escaneo de todos los puertos abiertos de la máquina objetivo, obteniendo como resultado, la información que se puede observar en la Fig. 14.

8.3.2. Ataques de diccionario

En este ataque el objetivo es obtener las credenciales del servicio de base de datos, en este caso, de una base de datos PostgreSQL. Tras realizar un primer escaneo de vulnerabilidades con Nessus, se obtiene que en la máquina objetivo tiene una vulnerabilidad en el puerto tcp 5432. Dicha vulnerabilidad, tiene el número de identificación CVE-2007-3280. Se informa que las credenciales utilizadas para

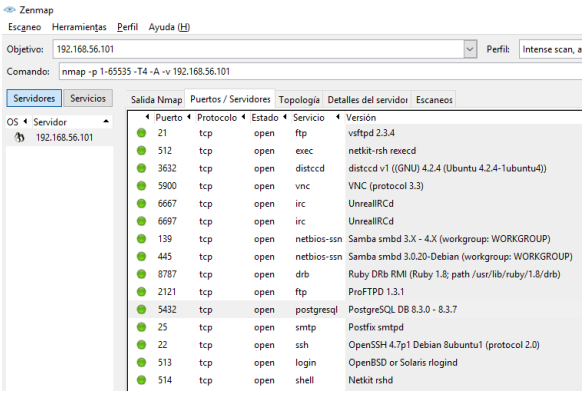

Fig. 14: Obtención del listado de los puertos abiertos de la máquina objetivo.

PostgreSQL son por defecto. Mediante el uso del framework Metasploit se realiza una búsqueda de esta vulnerabilidad hasta encontrar un exploit para explotarla. El exploit es *auxiliary/scanner/postgres/postgres_login*. Tras realizar las configuraciones pertinentes (indicar la dirección IP de la máquina objetivo) y como se puede observar en la Fig. 15, se obtienen las credenciales de la base de datos PostgreSQL.

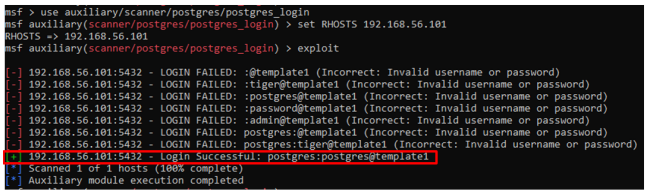

Fig. 15: Resultado del ataque realizado a la base de datos PostgresSQL.

8.3.3. Ataque al servicio vsftpd

En el escaneo de vulnerabilidades de la Fig. 14, se puede observar otra vulnerabilidad, en este caso, en el puerto 21. Este puerto tiene alojado el servicio ftp. Esta vulnerabilidad permite obtener una terminal en la máquina atacante mediante el uso de una conexión telnet. Al introducir el usuario finalizado con una cara sonriente ":)"se obtiene una terminal con privilegios de administrador. El exploit utilizado ha sido el *unix/ftp/vsftpd_234_backdoor*. El resultado del ataque se puede observar en la Fig. 16. Esta vulnerabilidad tiene el número de identificación CVE-2011-0762.

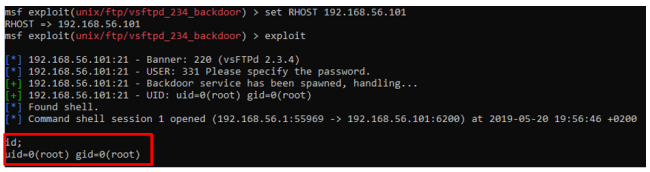

Fig. 16: Resultado del ataque realizado al servicio vsftp.

9 CONCLUSIONES

A lo largo el desarrollo del proyecto y de la elaboración de este documento, se ha mostrado la importancia de ga-

rantizar una mínima seguridad en el sistema, así como tener actualizados y de monitorizar los servicios que se utilizan. Nagios XI y Snort han tenido un papel importante es esta parte, ya que proporcionan en tiempo real, información constante de los equipos y sistema, para así poder actuar de la forma más eficiente para contrarrestar los posibles ataques que pueda sufrir las máquinas o el sistema.
Se ha analizado el framework Metasploit, así como la gran variedad de exploits que proporciona este framework para poder analizar en detalle el funcionamiento de estos, y la utilización de los mismos. Mediante el uso de Nessus, ha supuesto que la explotación de las vulnerabilidades se consiga realizar de forma eficiente. El uso de herramientas de escaneo de puertos como puede ser Nmap, ha servido para estudiar y analizar el primer paso que siguen la mayoría de los atacantes para obtener información de su objetivo.
A todo esto, hay que añadir las mesuras de seguridad implementadas mediante la implementación de firewalls con Iptables. Iptables proporciona una sintaxis muy básica, permite gestionar el tráfico de entrada y salida del equipo y/o red, y el funcionamiento concurrente con Snort, ha permitido capturar paquetes con comportamiento extraño y su posterior análisis.

10 Líneas futuras

En la realización del proyecto, por falta de tiempo y de no seguir una organización eficiente, no se han podido alcanzar todos los objetivos propuestos. Estos objetivos han sido la implementación de VLANs, implementación de backups en red, incluir una DMZ en la infraestructura de red actual e implementar medidas contra ataques de bots. Es por ello, que una continuidad del proyecto sería implementar los objetivos no alcanzados para incluir en la infraestructura actual de la red mucha mas seguridad. Por otro lado, estaría bien potenciar el uso de Snort, ya que a parte de detectar anomalías y monitorizar el tráfico, también proporciona protección y medidas frente a diferentes situaciones, que en mi parecer, sería interesante implementar en un futuro.

Agradecimientos

Agradezco a mi tutor del proyecto Juan Carlos Sebastián Pérez por haber confiado en mi para llevar a cabo el proyecto y por proporcionarme toda la ayuda posible para realizarlo.

Referencias

[1] ”Qué es SCRUM”, *Proyectos Ágiles*, https://proyectosagiles.org/que-es-scrum. 2018.

[2] *Lucidchart.com*, https://www.lucidchart.com.

[3] ”Common Open Research Emulator (CORE) — Networks and Communication Systems Branch”, *Nrl.navy.mil*, https://www.nrl.navy.mil/itd/ncs/products/core.

[4] A. Molina Coballes, ”Qué es iptables”, *OpenWebinars.net*, https://openwebinars.net/blog/que-es-iptables/. 2018.

[5] ”IPTables Red Hat Customer Portal”, *Red Hat Customer Portal*, https://access.redhat.com/documentation/es-es/red_hat_enterprise_linux/6/html/security_guide/sect-security_guide-iptables.

[6] I. Ramírez, ”¿Qué es una conexión VPN, para qué sirve y qué ventajas tiene?”, *Xataka.com*, https://www.xataka.com/seguridad/que-es-una-conexion-vpn-para-que-sirve-y-que-ventajas-tiene. 2018.

[7] ”IPsec: arquitectura de seguridad para IPv4 e IPv6”, *1&1 Digitalguide*, https://www.ionos.es/digitalguide/servidores/know-how/ipsec-arquitectura-de-seguridad-para-ipv4-e-ipv6/. 2016.

[8] ”¿En qué consiste una zona desmilitarizada (DMZ)?”, *1&1 Digitalguide*, https://www.ionos.es/digitalguide/servidores/seguridad/en-que-consiste-una-zona-desmilitarizada-dmz/. 2016.

[9] F. de Haro Bermejo, *Detección de intrusiones con Snort*, pp. 27-30. 2015.

[10] O. Sánchez Lorente, *DETECCIÓN DE INTRUSIONES CON SNORT*, PP. 11-13. 2015.

[11] ”Sistemas de Detección de intrusos y Snort”, *Maestros del Web*, http://www.maestrosdelweb.com/snort/. 2003.

[12] ”Nagios - Tenea tecnologias”, *Tenea.com*, https://www.tenea.com/servicios/nagios.html. 2016.

[13] M. Anicas, ”An Introduction to HAProxy and Load Balancing Concepts — DigitalOcean”, *Digitalocean.com*, https://www.digitalocean.com/community/tutorials/an-introduction-to-haproxy-and-load-balancing-concepts. 2014.

[14] I. Pérez, ”Que son y cómo funcionan los Buffer Overflow — WeLiveSecurity”, *WeLiveSecurity*, https://www.welivesecurity.com/la-es/2014/11/05/como-funcionan-buffer-overflow/. 2014.

[15] ”Vulnerability distribution of cve security vulnerabilities by types”, *Cvedetails.com*, https://www.cvedetails.com/vulnerabilities-by-types.php.

[16] H. Rizaldos, ”Qué es Metasploit”, *OpenWebinars.net*, https://openwebinars.net/blog/que-es-metasploit/. 2018.

[17] J. Martín, ”Nessus - Que es, como se usa”, *prezi.com*, https://prezi.com/e351d0eg5hsx/nessus-que-es-como-se-usa/. 2015.

[18] ”CVE -Common Vulnerabilities and Exposures (CVE)”, *Cve.mitre.org*, https://cve.mitre.org/. 2019.

[19] ”CVE -About CVE Entries”, *Cve.mitre.org*, https://cve.mitre.org/cve/identifiers/index.html#defined. 2019.