Universitat Oberta de Catalunya uoc.edu

Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones

SEGURIDAD EN LA INTERNET DE LAS COSAS

Autor: Nazaret García García-Maroto

Programa: Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones (MISTIC).

Consultor: Carlos Hernández Gañán
Profesor responsable de la asignatura: Victor García Font
Centro: Universitat Oberta de Catalunya

Junio 2018

Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada 3.0 España de Creative Commons

C) Copyright

© NAZARET GARCÍA GARCÍA-MAROTO
Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Seguridad en la Internet de las cosas*
Nombre del autor:	*Nazaret García García-Maroto*
Nombre del consultor/a:	*Carlos Hernández Gañan*
Nombre del PRA:	*Víctor García Font*
Fecha de entrega (mm/aaaa):	06/2018
Titulación:	*Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones*
Área del Trabajo Final:	*Área de ciberseguridad*
Idioma del trabajo:	*Español*
Palabras clave	*Seguridad, Redes, Raspberry Pi, Snort*

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

Debido a la evolución de las nuevas plataformas hardware que han surgido en el contexto de IoT (*Internet of Things*), se abre un gran camino para desarrollar proyectos que tienen que ver con el ámbito de la ciberseguridad y sus diferentes ramas ya que cada vez existen más dispositivos "inteligentes" conectados a Internet y eso conlleva a que se realice un especial hincapié en la seguridad perimetral de estos dispositivos con el fin de evitar que otras personas puedan controlar los dispositivos, obtener información sensible y confidencial, desconfigurarlos o incluso utilizarlos contra otros sistemas .

Por ello, el papel principal de este presente trabajo es presentar un proyecto de ciberseguridad cuyo objetivo es proporcionar un conocimiento teórico sobre la seguridad en general y específicamente en redes, un conocimiento práctico sobre cómo aumentar la seguridad perimetral y los datos que se transmiten por las redes y conseguir simular una red corporativa que puede ser usada en cualquier empresa de carácter medio y cuya implantación sea del menor coste posible.

Como se podrá ver a lo largo del proyecto, la plataforma elegida ha sido la Raspberry Pi 2 Modelo B y la solución elegida para aumentar la seguridad en la red de los usuarios es crear un servidor VPN gratuito con la placa elegida utilizando el protocolo OpenVPN debido a sus ventajosas características sin necesidad de utilizar

servidores costosos para proteger las conexiones de los dispositivos además de la configuración de Snort en la plataforma para poder analizar el tráfico de la red.

A lo largo del proyecto se irán viendo conceptos de instalación, puesta en marcha de la plataforma y configuraciones necesarias para el filtrado del tráfico de la empresa.

Abstract (in English, 250 words or less):

Due to the evolution of the new hardware platforms that have emerged in the context of IoT (Internet of Things), a great way is established towards the development of projects that have to do with the field of cybersecurity and its different branches, as a result of the increasing existance of "smart" devices connected to the Internet. That means that a special emphasis is placed on the perimeter security of these devices in order to prevent other people from controlling, obtain sensitive and confidential information, misconfigure or even use them against other systems.

Therefore, the main role of this paperwork is a cybersecurity project that has the objective of providing knowledge about general security and, specifically, on networks; a practical knowledge on how to increase perimeter security and the data transmitted by these networks; and get to simulate a corporate network that can be used in any other medium-sized company and whose implementation requires the lowest possible cost.

As you will be able to see throughout the project, the chosen platform has been the Raspberry Pi 2 Model B. The solution chosen to increase the security in the users network is to create a free VPN server with the chosen board using the OpenVPN protocol, as a result of its advanced features without the need of using pricey servers in order to protect the devices' connections, in addition to Snort's configuration so as to analize network traffic in the platform.

Throughout the project, concepts of installation, start-up of the platform and necessary adjustments for filtering the company's traffic will be shown.

Índice

1. Introducción 1
1.1. Contexto y justificación del Trabajo 1
1.2. Objetivos del Trabajo 2
1.3. Enfoque y método seguido 2
1.4. Planificación del Trabajo 3
14.1. Fase de recogida de información ……..3
1.4.2. Planificación del contenido del proyecto ……..3
1.5. Breve sumario de productos obtenidos 4
1.6. Breve descripción de los otros capítulos de la memoria 5
1.7. Riesgos y consumo de recursos 6
Capítulo 2: Seguridad perimetral y segmentación de redes 7
2.1. Introducción a la seguridad perimetral 7
2.1.1. Seguridad perimetral ……7
2.1.2. Segmentación de redes ……...10
2.2. Firewalls y zonas desmilitarizadas 13
2.2.1. Firewalls ……13
2.2.2. Zonas desmilitarizadas (DMZ) ……...16
2.3. Honeypots y honeynets 17
2.3.1. Honeypots ……...17
2.3.2. Honeynets ……18
2.4. IDS e IPS 18
2.4.1. IDS ……..19
2.4.2. IPS ……...19
2.5. Acceso remoto y redes virtuales 20
2.5.1. Funcionamiento de una VPN ……..20
2.5.2. Utilización de las conexiones VPN ……22
2.5.3. Ventajas e inconvenientes de las conexiones VPN ……22
Capítulo 3: Plataforma escogida para soluciones de seguridad en redes de bajo coste.. 23
3.1. Introducción a las plataformas de bajo coste 23
3.1.1. ARM ……23
3.1.2. Arduino ……...24
3.1.3. Raspberry Pi ……25

3.1.4. Otras plataformas ……28
3.2. Elección de la plataforma de bajo coste 28
3.2.1. ¿Qué se necesita? ……29
3.2.2. ¿Qué sistema operativo utilizar? ……31
Capítulo 4: Implementación de un servidor VPN en una plataforma Raspberry Pi..... 31
4.1. VPN..... 31
4.1.1. Protocolos para la implementación del servidor VPN ……32
4.2. Desarrollo de la solución elegida 33
4.2.1. Lado del servidor ……33
4.2.2. Lado del cliente ……...41
Capítulo 5: Snort y su implementación en una plataforma Raspberry Pi..... 44
5.1. Definición de Snort 44
5.2. Implementación de Snort 44
5.2.1. Instalación de la dependencia DAQ ……...45
5.2.2. Instalación de Snort ……46
5.2.3. Configuración y ejecución de Snort ……48
Capítulo 6: Validación del proyecto de seguridad..... 50
6.1. Utilización de Wireshark..... 50
6.1.1. Conexión sin VPN ……...50
6.1.2. Conexión con VPN ……51
6.2. Reglas de Snort..... 52
6.2.1. NMAP ……52
6.2.2. Inyección SQL ……...54
6.2.3. DoS ……55
6.2.4. Ataque de fuerza Bruta ……..55
6.2.5 Otras reglas ……..56
6.3. Validación del IDS Snort 57
Capítulo 7: Conclusiones..... 58
5.1. Conclusiones generales 58
5.2. Conclusiones específicas..... 58
5.3. Líneas de trabajo futuro..... 58
5.3.1. Capas de seguridad ……..59
5.3.2. Nuevas funcionalidades ……..59
Bibliografía..... 60

10. Anexos 62
Anexo A: Configuración y puesta en marcha de la Raspberry Pi 62
A.1 Configuración de aspectos iniciales ……...62
A.2. Conexión vía SSH a la Raspberry Pi ……64
Anexo B: Generar nombre de dominio dinámico. 67
Anexo C: Conexión con la VPN. 70

Lista de figuras

Figura 1.1. Planificación, diagrama de Gantt 4
Figura 2.1. Arquitectura sin seguridad perimetral. [1] 9
Figura.2.2. Arquitectura con seguridad perimetral. [1] 10
Figura.2.3. Red sin segmentar. [2] 11
Figura 2.4. Red con segmentación básica. [2] 14
Figura 2.5. Red con segmentación avanzada. [2] 14
Figura 2.6. Estructura de un sistema con DMZ y firewall. [3].....16
Figura 2.7. Localización potencial de honeypots en una red tradicional. [4] 18
Figura 4.1. Configuración del tráfico en un router ZTE.....35
Figura 4.2. Modificación del fichero vars. 35
Figura 4.3. Modificación del fichero vars 36
Figura 4.4. Configuración del certificado. 37
Figura 4.5. Configuración fichero sysctl.conf. 40
Figura 4.6. Configuración fichero interfaces. 40
Figura 5.1. Página principal de Snort.. 45
Figura 5.2. Versión de Snort instalada. 47
Figura 5.3. Modificación del fichero snort.conf. 48
Figura 5.4. Ejecución de la herramienta Snort... 49
Figura 6.1. Filtrado de paquetes en el sniffer Wireshark.... 51
Figura 6.2. Detalles de un paquete en el sniffer Wireshark.....51
Figura 6.3. Paquetes cifrados en el sniffer Wireshark.....52
Figura 6.4. Reglas de Snort.... 57
Figura 6.5. Regla “Escaneo NMAP UPD” 57
Figura 6.6. Regla “Escaneo NMAP Ping” 57
Figura A.1. Configuración del fichero interfaces 65
Figura A.2. IP fija de la Raspberry Pi.....66
Figura B.1: Sign Up en No-IP 67
Figura B.2: Panel de control del usuario en No-IP. 68
Figura B.3: Creación de host en No-IP 68
Figura C.1: Cuadro de diálogo OpenVPN GUI.....70
Figura C.2: Conexión OpenVPN GUI. 70

1. Introducción

1.1. Contexto y justificación del Trabajo

En el presente Trabajo Fin de Máster (TFM), se va a presentar un proyecto de ciberseguridad cuyo objetivo es conseguir simular una red corporativa que puede ser usada en cualquier empresa de carácter medio y cuya implantación sea del menor coste posible.

Tras la comparación de diferentes plataformas que se pueden utilizar para este tipo de proyectos, se ha elegido la placa Raspberry Pi 2 Modelo B debido a sus características tanto en hardware como en software que son más que suficientes para realizar este proyecto, además de por su precio y facilidad para adquirir conocimientos prácticos sobre informática, y en concreto en este proyecto, sobre seguridad.

A lo largo del proyecto se analizará la plataforma de una manera más detallada, así como ejemplos de iniciación de la misma de una manera didáctica utilizando una solución open-source además de la configuración de Snort en la plataforma para poder obtener el tráfico de la red.

El objetivo de este trabajo, cómo se ha mencionado, es proporcionar un conocimiento teórico sobre la seguridad en general y específicamente en redes, y un conocimiento práctico sobre cómo aumentar la seguridad perimetral y los datos que se transmiten por las redes. La idea final del proyecto es conseguir un sistema reducido que sea capaz de realizar las tareas, a menor escala, que un sistema de seguridad completo en una empresa, de modo que pueda servir a las empresas para hacer una demostración del servicio.

A lo largo del proyecto se irán viendo conceptos de instalación, puesta en marcha de la plataforma y configuraciones necesarias para el filtrado del tráfico de la empresa.

En cuanto al contexto del Trabajo Fin de Máster, se desarrolla en el año 2018, un año en el que se siguen produciendo una gran cantidad de ciberataques en todo el mundo lo que hace que la seguridad Informática esté a la orden de día.

Se viven unos años en los cuales las tecnologías, y lo que se denomina el *Internet of Things* (el Internet de las cosas), está muy presente en el día a día de las personas. La mayoría de los usuarios poseen como mínimo un dispositivo electrónico, como puede ser un ordenador, un teléfono inteligente, una tableta, un reloj inteligente…

existen multitud de dispositivos electrónicos al alcance de los usuarios que permiten hacer la vida más sencilla. Sin embargo, estos dispositivos electrónicos están conectados a Internet, lo que les hace vulnerables a los ciberataques si no cuentan con un mínimo de seguridad.

Se producen ataques que atentan contra la seguridad de los datos de los usuarios tanto a particulares como empresas ya que hoy en día los datos son el nuevo petróleo del sigo XXI y el objetivo es obtener información crítica.

Por ello, hoy más que nunca la seguridad Informática es un pilar fundamental en la sociedad y debe estar presente en las tecnologías que utilicen los usuarios.

1.2. Objetivos del Trabajo

Como se ha comentado en la introducción del propio *Capítulo 1: Introducción y objetivos del trabajo*, el objetivo del proyecto es conseguir simular una red corporativa que puede ser usada en cualquier empresa de carácter medio, de modo que de alguna forma su implantación sea menos costosa. Para ello lo que se propone es la simulación de una red de empresas dentro de una Raspberry PI utilizando una solución open-source.

A continuación, se listan una serie de objetivos para conseguir llegar a este objetivo final:

- Crear una VPN en una plataforma con el menor coste posible. Esto permite a cualquier posible empleado de esta empresa entrar a los recursos de la red de esta de manera cifrada y segura.
- Montaje de un IDS/IPS con el menor coste posible para poder monitorizar el tráfico de la red de la empresa.
- Crear un punto de unión entre la VPN y el IDS para conseguir una red protegida.

Llevando a cabo los anteriores objetivos, se llegará al objetivo final del proyecto.

1.3. Enfoque y método seguido

La estrategia seguida para llevar a cabo este proyecto ha sido utilitzar una plataforma ya existente, la plataforma Raspberry Pi, en ella se ha adaptado la implementación de una VPN y se ha instalado la herramienta Snort. Esta estrategia ha sido utilitzada debido a que se buscaba securizar una red de la manera menos costosa

posible y que fuese eficaz. Con la implementación realizada se cumplen los objetivos sin necesidad de utilizar plataformas o herramientas más costosas o de pago.

1.4. Planificación del Trabajo

Este trabajo forma parte del trabajo de fin de máster de la titulación Máster en seguridad de las tecnologías de la información y de las comunicaciones, y ha sido desarrollado en diferentes fases, las cuales se presentan a continuación.

14.1. Fase de recogida de información

La primera fase para iniciar un proyecto es la fase de recogida de información acerca del tema que se va a investigar.

Lo primero que se realizó fue investigar sobre el tema del proyecto, la seguridad en las redes, y buscar que soluciones se podían implementar.

Tras haber realizado lo anterior, se buscaba una plataforma que pudiese realizar proyectos relacionados con la seguridad por las características de la misma y sobre todo que fuese de bajo coste para que cualquier empresa de carácter medio pudiese permitirse adquirirla.

La plataforma que reunía todas estas características es la plataforma Raspberry Pi, debido a su calidad/precio es idónea para este proyecto además de ser una de las más famosas.

Por otro lado, se buscaba una herramienta compatible con la plataforma escogida que permitiese detectar y monitorizar el tráfico de la red. La herramienta de seguridad elegida y compatible es Snort, cuenta con un gran potencial de análisis con la implementación de reglas de detección que permiten controlar aquellos paquetes de interés.

1.4.2. Planificación del contenido del proyecto

Para conseguir alcanzar estos objetivos marcados, se siguieron una serie de tareas a realizar, de modo que si se cumplen estas tareas se cumpla el objetivo final.

La primera tarea consistió en analizar y estudiar el mercado de la ciberseguridad, observar las últimas noticias en este ámbito y buscar una solución para poder acceder a las redes de una manera segura.

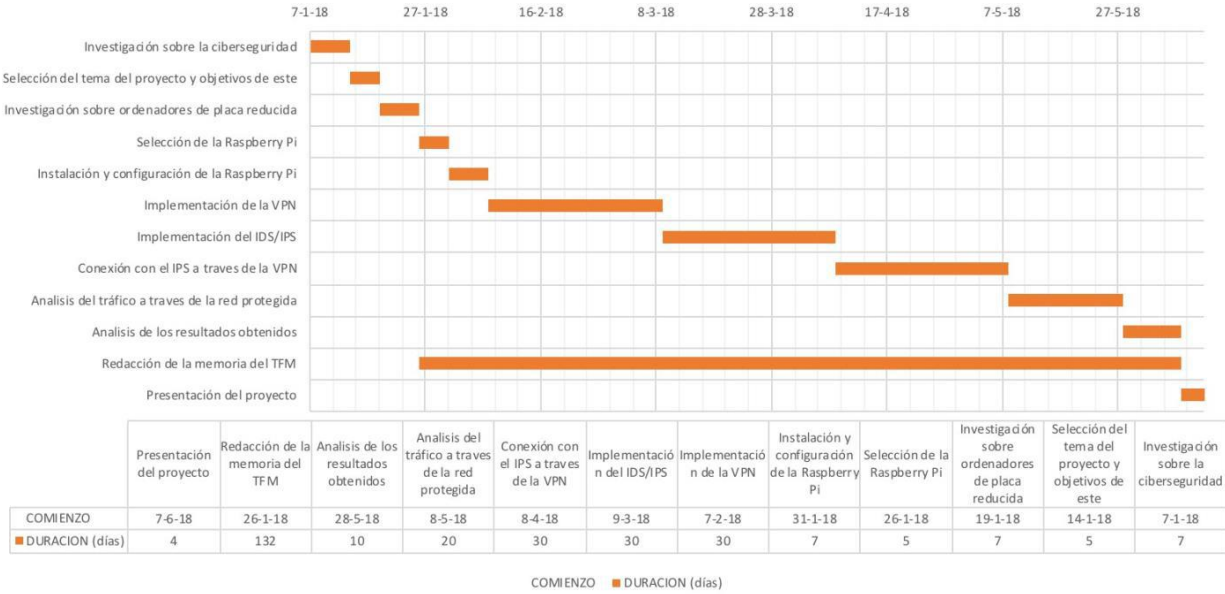
La segunda tarea consistió en investigar sobre diferentes ordenadores de placa reducida y seleccionar uno de ellos en función de sus características económicas y hardware/software.

La tercera tarea consistió en instalar y configurar la placa elegida e implementar la VPN con un software open-source, en este caso OpenVPN.

La cuarta tarea consistió en configurar e implementar el IDS/IPS elegido, en este caso mediante reglas de Snort.

La quinta tarea consistió en conectar con el IDS/IPS a través de la VPN y analizar el tráfico de la red protegida. Y, por último, estudiar los resultados y obtener una conclusión sobre el grado de seguridad y detección de anomalías en el tráfico analizado.

Tras esta tarea, se empezó a realizar la documentación de la parte práctica del proyecto, que duró las siguientes dos semanas.



	Presentación del proyecto	Redacción de la memoria del TFM	Analisis de los resultados obtenidos	Analisis del tráfico a traves de la red protegida	Conexión con el IPS a traves de la VPN	Implementación del IDS/IPS	Implementación de la VPN	Instalación y configuración de la Raspberry Pi	Selección de la Raspberry Pi	Investigación sobre ordenadores de placa reducida	Selección del tema del proyecto y objetivos de este	Investigación sobre la ciberseguridad
COMIENZO	7-6-18	26-1-18	28-5-18	8-5-18	8-4-18	9-3-18	7-2-18	31-1-18	26-1-18	19-1-18	14-1-18	7-1-18
DURACION (días)	4	132	10	20	30	30	30	7	5	7	5	7

Figura 1.1. Planificación, diagrama de Gantt.

La planificación de los tiempos comentados en este apartado pueden observarse en la *Figura 1.1*. Esta figura es un diagrama de Gantt en el cual se muestra la división del proyecto en tareas, y el tiempo en el que se llevaron a cabo cada una, expresado en días.

1.5. Breve sumario de productos obtenidos

Los productos obtenidos del presente trabajo se describen a continuación, y la descripción detallada se realizará en los siguientes apartados.

- Memoria del documento. Esto consiste en este documento que detalla paso a paso los procedimientos seguidos, su resultado y su valoración a futuro.

- Guía de implementación de una Red Privada Virtual en la plataforma Raspberry Pi.
- Guía de implementación de Snort en la plataforma Raspberry PI.
- Reglas de detección de ataques en Snort.
- Presentación de resultados.

1.6. Breve descripción de los otros capítulos de la memoria

En el segundo capítulo, se desarrollan los conocimientos teóricos necesarios para entender el trabajo, y completar todos los conocimientos sobre la seguridad Informática. Este segundo capítulo, se divide en varias secciones en las cuales en un primer lugar se introduce una idea de lo que es la seguridad perimetral y conceptos relacionados, y se continúa explicando diferentes dispositivos como soluciones para fomentar la seguridad del perímetro y la segmentación de redes. Este capítulo es completamente teórico, en él se desarrollan diversas definiciones y conceptos que han sido extraídos de la web, libros o documentos que se referencian en la bibliografía del trabajo.

El tercer capítulo se centra en el análisis de las diferentes plataformas escogidas, se explican las características principales, ventajas y usabilidad de las mismas, pero en profundidad se detalla por qué la placa Raspberry Pi es la plataforma elegida. El objetivo es que el lector obtenga un conocimiento completo de la placa tanto en su parte hardware como software, por ello se describen los materiales necesarios para comenzar el proyecto, la instalación y configuración inicial para ponerla en funcionamiento.

El cuarto capítulo, se compone del grueso en cuanto a programación y conocimientos de este proyecto, este se centra en la creación de un servidor VPN y en el que se explica paso a paso y con ayuda de apéndices la implementación del mismo. Como se ha mencionado con anterioridad, en este capítulo es necesario desenvolverse bien con el sistema operativo Linux.

En el quinto capítulo, se explica qué es la herramienta Snort, cual es su finalidad y la implementación de la misma en la plataforma Raspberry Pi.

En el sexto capítulo, se incluyen las pruebas de validación del servidor mediante el sniffer Wireshark para demostrar que el tráfico que pasa por el VPN está cifrado y se desarrollan las reglas creadas para la herramienta Snort que detectan cualquier anomalía en el tráfico de la red.

En el séptimo capítulo, se describen las conclusiones tras haber realizado el proyecto, lo que se ha aprendido en la fase de investigación tras haber profundizado en

el tema de seguridad y redes y la parte de implementación de la misma, además se desarrollan líneas de trabajo futuro.

En el octavo capítulo se encuentra la bibliografía que permite al lector ampliar información sobre los conceptos que aparecen en el trabajo, así como diversos apéndices, que servirán como complemento.

En el último capíuto, se han añadido Anexos complementarios al proyecto que explican la configuración puesta en marcha de la plataforma Raspberry Pi, la creación de dominios dinámicos y la conexión con la VPN.

1.7. Riesgos y consumo de recursos

Se van a analizar los riesgos que se pueden encontrar en cuanto a la implementación de la plataforma, así como posibles complicaciones que podemos encontrar. Se dividen los riesgos en dos tipos:

- Riesgos de seguridad. Se debe encontrar una plataforma que cumpla con las características hardware y software necesarias para el desarrollo del proyecto y además sea segura. Es por ello por lo que una configuración correcta de la placa y del IDS/IPS elegido es vital para evitar riesgos de seguridad y detectar posibles ataques a la red.
- Riesgos económicos. En cuanto al apartado económico, el único problema encontrado es un supuesto calentamiento excesivo de la placa, sin embargo, si se utilizan los disipadores necesarios como prevención, es algo relativamente imposible que pase. Además, se ofrece una red protegida por menos de 50€ ya que una placa Raspberry Pi no supera los 30€ en el mercado y el resto de los costes irían ligados a elementos hardware como la fuente de alimentación o la tarjeta SD.

En cuanto a consumo de recursos, RPi-Monitor es una aplicación que permite monitorizar ciertos parámetros de la placa Raspberry Pi como la carga de la CPU, el consumo de memoria RAM y la temperatura del procesador en tiempo real. En general y en concreto en este proyecto, el consumo de recursos por parte de a plataforma es escaso y permite una gran escalabilidad en todo tipo de proyectos.

Capítulo 2: Seguridad perimetral y segmentación de redes

2.1. Introducción a la seguridad perimetral

En esta primera sección del capítulo dos, se va a hablar sobre la seguridad perimetral y sobre la importancia de la segmentación de las redes, además de comentar diferentes dispositivos que contribuyen a esta seguridad.

2.1.1. Seguridad perimetral

La seguridad Informática clásica se centraba en la prevención, basada casi por completo en la protección del perímetro. El objetivo de esta seguridad era evitar ataques desde el exterior, salida de información desde el interior y accesos no autorizados, por lo que el perímetro, en resumen, está formado por un conjunto de máquinas y dispositivos que se encuentran en la frontera de nuestra red para proteger perímetros internos y externos.

Al principio, la entrada y salida de datos se controlaba con métodos de seguridad física pero debido a que la tecnología ha ido avanzando a lo largo de los años, la seguridad física ha resultado exigua y por lo tanto ha llevado a la utilización de medidas lógicas para permitir o no los accesos.

Debido a esta evolución de la tecnología, el acceso a la red desde el exterior se puede realizar de manera inalámbrica, mediante dispositivos móviles, acceso remoto... aumentando el acceso a arquitecturas SaaS y redes sociales, que conlleva a que las políticas basadas en puertos y protocolos, habitualmente utilizadas para la seguridad perimetral, sean menos seguras. Además, el control de las comunicaciones a través del navegador cada vez es más complicado debido a que el contenido puede ir cifrado y existen diversas aplicaciones web. Por esto, se necesitan de nuevas herramientas, máquinas o dispositivos para afrontar el progreso de los usuarios con la tecnología, para contrarrestar los nuevos mecanismos de ataque contra los sistemas de información y comprender las diferentes maneras en las que los procesos de negocio incorporan las tecnologías de la información y comunicación.

Debido a lo anterior se crean nuevas herramientas para proteger el perímetro, y según el Instituto Nacional de Ciberseguridad [5] se diferencian dos tipos:

- Cortafuegos, VPN (Virtual Private Network) e IPS/IDS (Intrusion Prevention Systems/ Intrusion Detection Systems).

- Gestión y control de acceso e identidad, en particular la subcategoría de Control de acceso a red.

Ambas herramientas nos protegen tanto de amenazas externas relacionados con accesos no autorizados procedentes de la red o de servicios web como de amenazas internas impidiendo que la información del usuario se envíe.

Y permiten:

- Ocultar la dirección IP y puertos de los equipos de nuestra red a los intrusos y a los que intentan conocer la estructura de nuestra red.
- Monitorear las comunicaciones entre redes impidiendo la saturación de servicios (ataques DoS (Denial Of Service) y DDoS (Distributed Denial of Service)).
- Conectarnos a redes privadas con seguridad.
- Saltar posibles bloqueos por país.
- Evitar boicots de los ISP (Internet Service Provider) al uso de redes P2P (Peer-To-Peer).

Por otro lado, hay que destacar que estos dispositivos no protegen de aquellos ataques cuyo tráfico no pase por ellos, de ataques de ingeniería social o de virus informáticos que se encuentran en archivos o software.

En resumen, estas herramientas aumentan la seguridad protegiéndonos del exterior y las siguientes secciones se centrarán más profundamente en estas herramientas.

2.1.1.1. Importancia de una arquitectura con seguridad perimetral

Existen diferentes dispositivos que se encuentran en la frontera de la red como por ejemplo routers frontera, cortafuegos (firewalls), sistemas de detección de intrusos (IDS), redes privadas virtuales (VPN), software y servicios como Bastion Host que es un sistema informático que ha sido fuertemente protegido para soportar ataques desde un lugar hostil (Internet) o zonas desmilitarizadas (DMZ).

Debido a la importancia de asegurar el perímetro de una red, se muestra la diferencia entre una arquitectura sin seguridad perimetral y una arquitectura con seguridad perimetral.

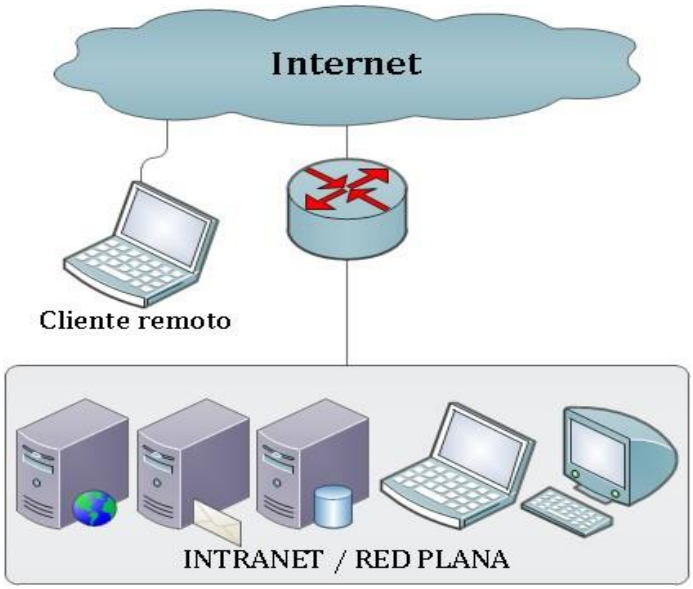


Figura 2.1. Arquitectura sin seguridad perimetral. [1]

En la *Figura 2.1*, se muestra una red sin segmentar, no existen elementos de monitorización por lo que no se puede administrar la red, no se filtran datos ni de entrada ni de salida por lo que no se verifica las amenazas externas de la red quedando la misma totalmente expuesta a los riesgos y no se controla la salida de información comprometiendo así información confidencial entre otras.

En la *Figura 2.2*, se muestra todo lo contrario, existen herramientas o dispositivos que protegen el perímetro como firewalls y DMZ con su correspondiente antispam y antivirus permitiendo filtrar los datos de entrada y salida, la red esta segmentada aumentando así la seguridad y dando por ello lugar a la defensa en profundidad (prevención) y al principio de mínimo privilegio además de reducir el tráfico.

Las máquinas y dispositivos, comentados anteriormente, que se encuentran en la frontera de nuestra red, actúan como primera línea de defensa.

Como por ejemplo los Routers frontera que realizan el primer filtrado de paquetes utilizando ACL (Acces Control List), que son una serie de reglas para establecer que tráfico se permite y que tráfico se deniega, o los firewalls, que son hardware o software dedicados específicamente a esta tarea de control e inspección de paquetes que pasan por él, para decidir si prosiguen o no su camino. El firewall más simple es el filtro de paquetes que sigue el esquema de lista de reglas ACL como el router frontera anterior, inspeccionando las cabeceras de los paquetes a nivel de red

(puerto, direcciones IP's...) para decidir si pasan o no a nuestra red. Más adelante se profundizará en los diferentes firewalls existentes. [1]

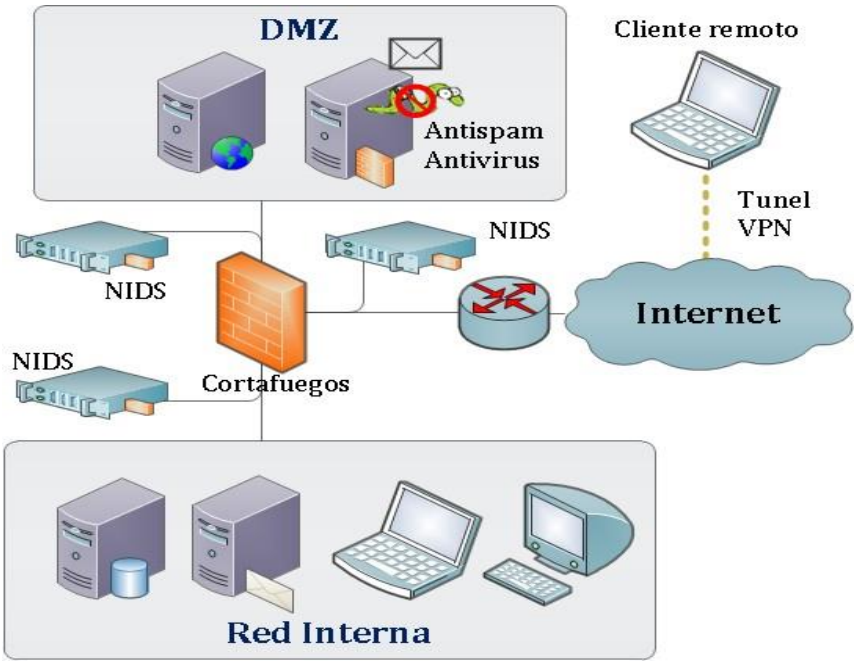


Figura.2.2. Arquitectura con seguridad perimetral. [1]

2.1.2. Segmentación de redes

Como se ha visto en la sección *2.1.1. Seguridad perimetral*, es muy importante asegurar lo máximo posible el perímetro, sin embargo, no es suficiente con esto, ya que, si un atacante logra acceder, toda la red quedará expuesta. Por ello, también es muy importante segmentar la red del sistema ya que, si un atacante decide actuar sobre la red, se añade mayor dificultad para realizar el ataque.

Además de aumentar la seguridad de la red y aumentar el ancho de banda por usuario, segmentarla permite establecer políticas de seguridad más adecuadas de acceso a la red limitando el acceso a los usuarios, ya que todos no estarán autorizados para acceder a las mismas zonas de la red. Para ello, necesitamos dispositivos que puedan ser configurables y modulares.

2.1.2.1. Redes no segmentadas

Una red no segmentada da lugar a un menor ancho de banda para los usuarios de la red, se producen cuellos de botella, ya que no se aísla el tráfico y se producen colisiones y como no, una menor seguridad ya que si un atacante consigue entrar en ella, tendrá acceso total a los servidores de la red al tratarse de una red plana.

Según un estudio de la empresa Vera Quest Research, solicitado por la empresa Avaya Networking (14 septiembre 2016), muy pocas empresas a estas alturas de la evolución de las tecnologías y el aumento de los ciberataques tienen una segmentación completa de las redes. El estudio reveló que "Sólo 1 de cada 4 profesionales TIC afirma contar con este despliegue y hasta un 22% incluso desconocía que puede hacerse", de acuerdo con un posterior estudio. [6]

En la *Figura 2.3*. se muestra una red tradicional, sin segmentar, en la que todas las redes están conectadas entre sí y no existen elementos de seguridad para separarlas.

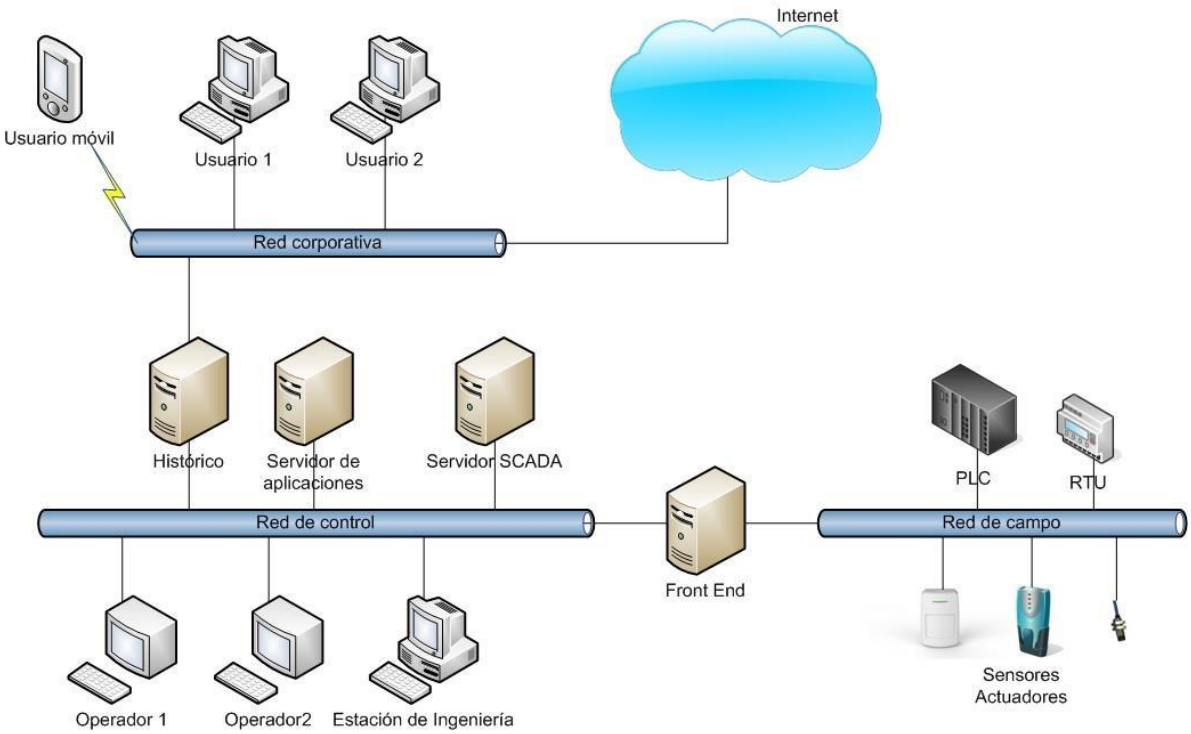


Figura.2.3. Red sin segmentar. [2]

2.1.2.2. Redes segmentadas

Como se ha comentado en la introducción de esta sección, se necesitan dispositivos que puedan ser configurables para establecer políticas de seguridad y modulares por si fuese necesario ampliar la red en un futuro.

Existen diferentes dispositivos de electrónica de red, comentados anteriormente, que permiten segmentar las redes. Dos de los más destacados son:

- Switches: Permiten dividir la LAN en partes más pequeñas, más eficaces y fáciles de administrar utilizando la topología Ethernet que permite utilizar de

manera más eficiente el ancho de banda pudiendo incrementarlo más que otros dispositivos y por ello descongestionar el tráfico.

- Firewalls: Es un dispositivo de seguridad hardware/software que tiene como objetivo proteger una red de las intrusiones de otras redes a las cuales está conectada (por ejemplo, Internet).

Con estos dispositivos se puede conseguir la segmentación de la red de diferentes maneras:

- VLANs: La manera más sencilla de segmentar una red es mediante redes de área local virtuales (VLANs), que permiten mejorar el rendimiento al separar los protocolos de las redes, ya que se reduce el tráfico innecesario de la red, segmentar el dominio reduciendo así la cantidad de dispositivos por dominio y por ello el área de visibilidad de cada zona y aumentar la seguridad ya que a cada grupo de usuarios se le asigna una determinada VLAN permitiendo así que su comunicación este separada por completo del tráfico de datos del resto de grupos. [7]
- Whitelisting/Blacklisting: Son técnicas de segmentación que permiten garantizar la seguridad de la red. Whitelisting controla lo que debe permitir acceder a la red mediante un análisis de los paquetes entrantes, mientras que Blacklisting se centra en lo que debe ser denegado. La técnica Whitelisting es más vulnerable comparada con la técnica Blacklisting siempre que sea factible su implementación.
- DMZ (Zona Desmilitarizada): Es una zona segura que se sitúa entre la red interna de una organización y una red externa, por lo que se plantea como un paso intermedio entre nuestra red y el acceso a Internet. Se comentará en más detalle en la sección *2.2.2. DMZ.*

2.1.2.3. Segmentación básica

En la *Figura 2.4.* se muestra una red segmentada de forma básica, en la que las redes están separadas por firewalls en este caso, pero también podrían estar separadas por switches como se explicaba anteriormente. Debido a que son dispositivos configurables, se le pueden aplicar una serie de políticas de seguridad para el control de salida y entrada de datos. [2]

2.1.2.4. Segmentación avanzada

En la *Figura 2.5* se muestra una red segmentada de forma avanzada, además de contar con dispositivos de seguridad comentados en la red segmentada básica, añade diodos de datos y dispositivos IPS. También se encuentra en la arquitectura un DMZ, que se explicará más adelante, para separar la Red corporativa de la Red de datos protegiendo así la red interna de la red externa. [2]

Una solución propuesta por varias empresas para la segmentación de las redes de inicio a fin incluye tres funcionalidades:

- Hípersegmentación que es la capacidad de crear segmentos ocultos que cruzan la red entera.
- Ocultamiento nativo que es la característica de un hípersegmento que es invisible para los piratas informáticos.
- Elasticidad automatizada que permite crear y eliminar automáticamente hípersegmentos.” [8]

2.2. Firewalls y zonas desmilitarizadas

2.2.1. Firewalls

Un firewall es un dispositivo de seguridad hardware/software que tiene como objetivo proteger una red de las intrusiones de otras redes a las cuales está conectada (por ejemplo, Internet). Decide si debe permitir o denegar las transmisiones de una red a otra en función de un conjunto de políticas de seguridad definidas en la red que deben proteger.

Por ello, es de vital importancia la localización del firewall, debe situarse en la zona de la red donde se recoja todo el tráfico entrante y saliente para su monitorización.

La funcionalidad principal de un firewall es controlar el tráfico que pasa por él, aplicar a este tráfico una serie de filtros en busca de patrones y una vez encontrados estos patrones, aplicar reglas de filtrado para operar con ellos.

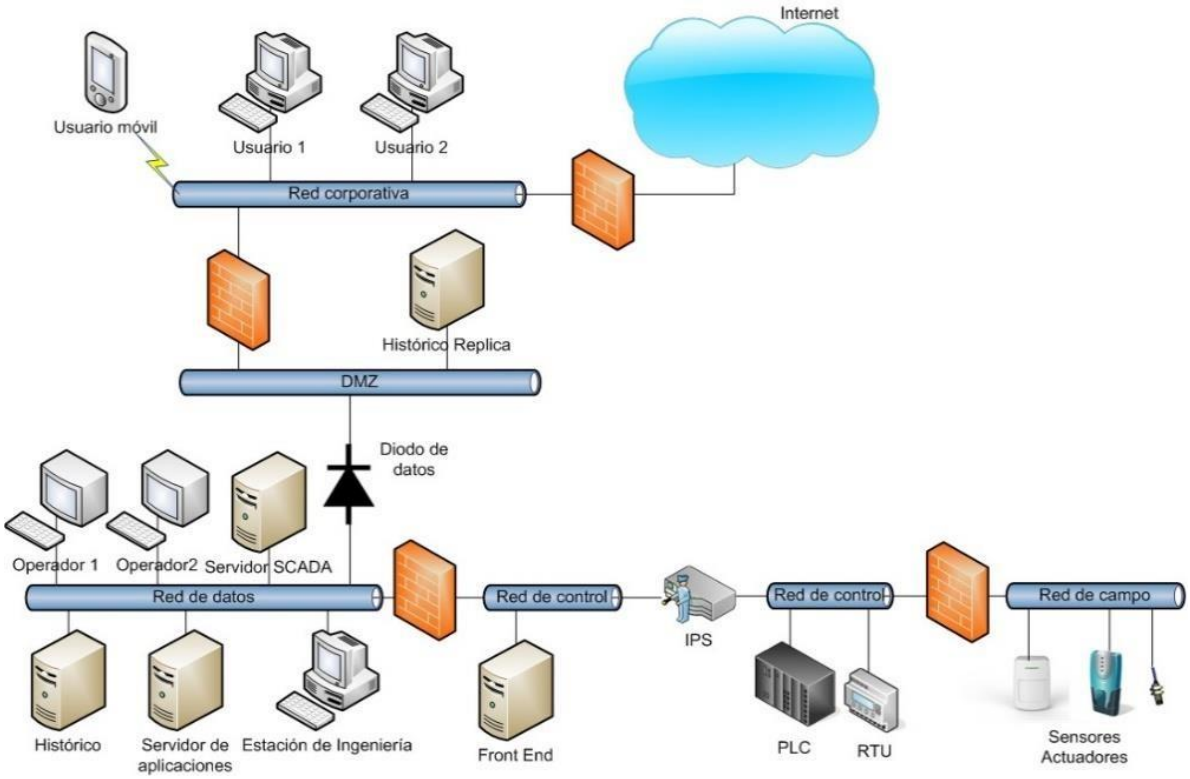


Figura 2.4. Red con segmentación básica. [2]

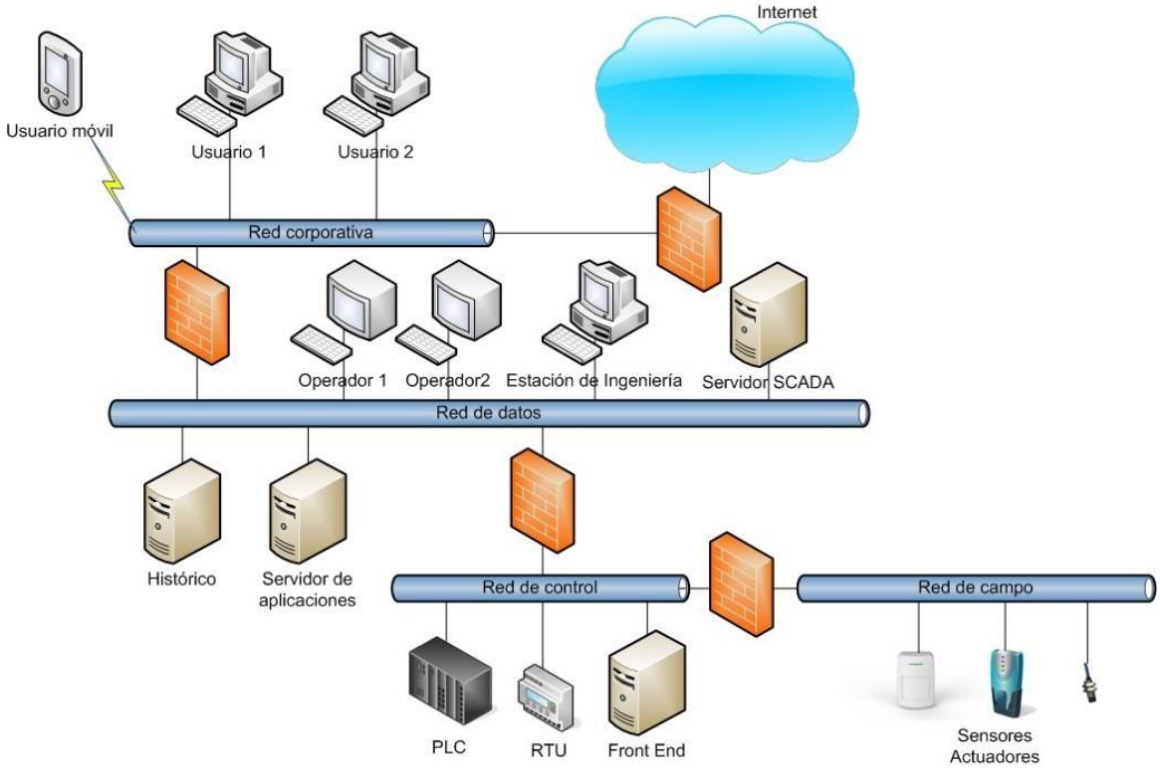


Figura 2.5. Red con segmentación avanzada. [2]

Los firewalls se pueden clasificar atendiendo a su nivel de funcionamiento [9]:

- Firewall a nivel de red/transporte: Controlan las comunicaciones entre redes a nivel de capa de red, examinando las cabeceras de cada paquete para permitir o denegar el tráfico y filtrar por puerto, dirección, etc. mediante routers o enrutados.
- Firewalls DPI (Deep Packet Inspection): Permiten filtrar por protocolos y contenidos específicos, como por ejemplo SOAP (Simple Object Access Protocol) o XML (eXtensible Markup Language).
- Firewall a nivel de aplicación (proxy): Controlan las comunicaciones entre redes a nivel de capa de aplicación, realiza un filtrado y seguimiento más concreto de las solicitudes y respuestas utilizando protocolos específicos de los parámetros de cada aplicación como por ejemplo HTTP (Hypertext Transfer Protocol), SMTP (Simple Mail Transfer Protocol), FTP (File Transfer Protocol).

Existe otra clasificación de los firewalls en función de las reglas de filtrado definidas. [9]

- Políticas permisivas o restrictivas.

O pueden clasificarse por el tipo de activo que protegen.

- El firewall de perímetro o red (hardware) o el firewall de sistema o nodo (software).

Aunque los firewalls se utilicen como dispositivos de seguridad tienen algunas limitaciones ya que sólo sirven de defensa perimetral [10].

Las limitaciones más características son que solo protegen frente al tráfico que atraviesan el firewall, por lo tanto, no pueden actuar contra todos los ataques que pueda recibir nuestra red, uno no es suficiente ya que se necesitan de más dispositivos redundantes por si el firewall pudiese fallar o el atacante lograse burlarlo, se debe complementar con software antivirus, si el firewall no tiene una correcta configuración puede no proteger de los fallos de seguridad de los servicios y protocolos de los cuales se permita el tráfico, y además no tienen nada que hacer contra la técnica de Ingeniería social, que es una técnica utilizada para la recogida de información valiosa a partir de la manipulación psicológica de las personas con el fin de realizar un ataque.

2.2.2. Zonas desmilitarizadas (DMZ)

Una Zona Desmilitarizada (DMZ) se trata de una zona segura que se sitúa entre la red interna de una organización y una red externa, por lo que se plantea como un paso intermedio entre nuestra red y el acceso a Internet (*Figura 2.6*).

La zona desmilitarizada permite que servidores interiores puedan dar servicios a la red exterior a la vez que protegen la red interior en el caso de que intrusos comprometan la seguridad de los equipos situados en la zona desmilitarizada.

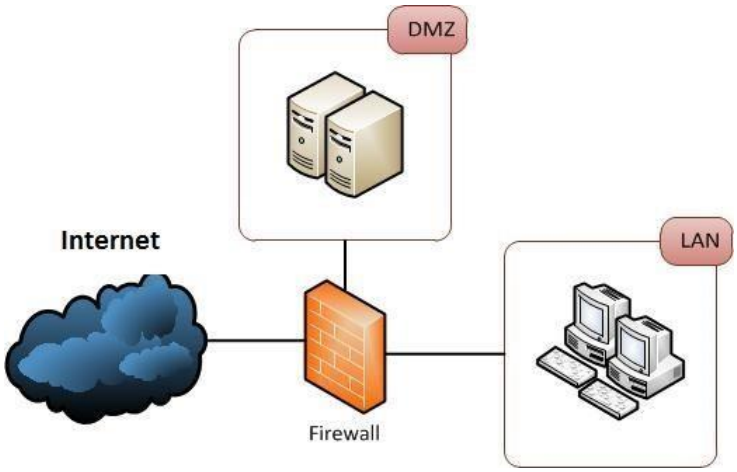


Figura 2.6. Estructura de un sistema con DMZ y firewall. [3]

La DMZ puede ser más segura si se implementa con un firewall de tres vías (DMZ protegida) o con dos firewalls, siendo esta última la opción más segura y por lo tanto más costosa.

Si se utilizan dos firewalls, un firewall separa el exterior de la red del segmento desmilitarizado (la DMZ) y los servidores que tienen que ser públicos desde el exterior de la red. El segundo firewall, que hace de punto de contacto entre la red interna y la zona desmilitarizada, se configurara para que rechace todos los intentos de conexión que vayan llegando desde el exterior. “Así, si un atacante consigue introducirse en uno de los servidores de la zona desmilitarizada, será incapaz de atacar inmediatamente una estación de trabajo. Es decir, aunque un atacante se apodere del segmento de los servidores, el resto de la red continuara estando protegida mediante el segundo de los cortafuegos.” [11]

2.3. Honeypots y honeynets

Debido al incremento de ataques informáticos en los últimos años, se han ido creando nuevas herramientas para aumentar la seguridad en las redes. Dos herramientas que permiten obtener el origen de los ataques son las honeypots y honeynets.

2.3.1. Honeypots

Una honeypot "tarro de miel" es un recurso relacionado con la seguridad Informática que simula ser un sistema vulnerable con el objetivo de ser puesto a prueba, atacado y/o comprometido para atraer atacantes y recoger información sobre tácticas y herramientas utilizadas por los mismos. Al ser sistemas pasivos, no reactivos, son más de detección que de proporcionar respuesta ya que no tienen capacidad de reaccionar.
Suelen ser herramientas muy útiles para aprender acerca de los ataques zero-day e incluso para detectarlos a tiempo. Un ataque de zero-day se aprovecha de una vulnerabilidad que muy pocas personas conocen y que por lo tanto todavía no ha sido resuelta.

A la hora de utilizar un honeypot hay que tener en cuenta [9] que se necesita que el honeypot sea atacado, tiene que ser un sistema lo más realista posible conectado a una red real, por lo que se tendrá que generar para él, datos y procesos falsos y cambiar las configuraciones por defecto, se debe separar el honeypot adecuadamente del resto de sistemas para que no se vean comprometidos (*Figura 2.7*), en función de cuál sea el objetivo se dejará un poco más desprotegido o no el honeypot. Y por supuesto nadie de la organización debe acceder al honeypot ya que todo lo que llegue a este sistema se clasificará como un potencial ataque.

El resumen de esta herramienta es que los beneficios que otorga son una alta precisión de detección del ataque o atacante, consume una gran cantidad de tiempo del atacante para realizar el ataque, es muy eficaz si se emplea correctamente y se puede utilizar en diferentes propósitos. Los contras de esta herramienta por otro lado es que es difícil de manejar y los atacantes experimentados han aprendido como ignorar este tipo de herramientas dando lugar a un sistema vulnerable poniendo en peligro los recursos de la organización.

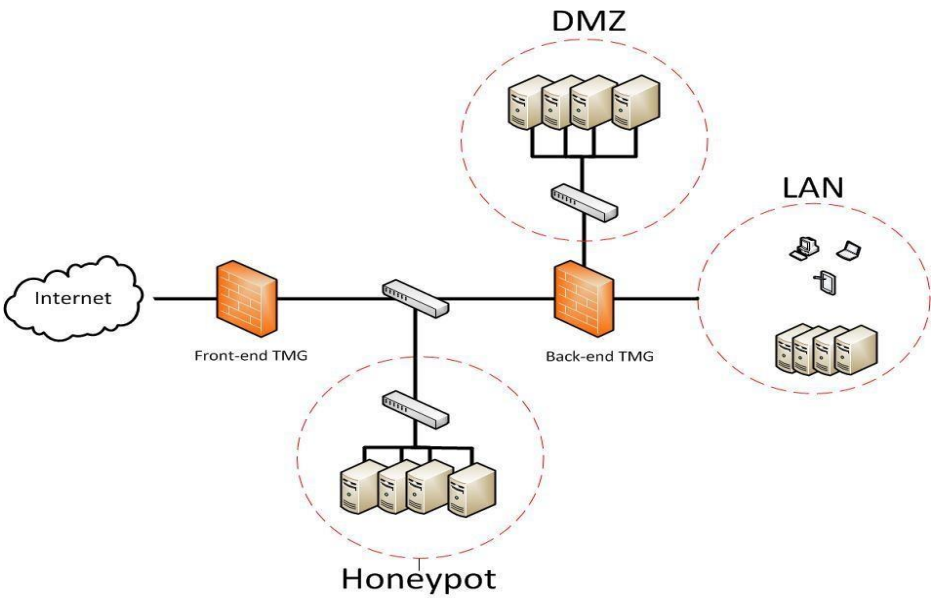


Figura 2.7. Localización potencial de honeypots en una red tradicional. [4]

2.3.2. Honeynets

Una honeynet es un conjunto de honeypots de alta interacción que conforman una red completa permitiendo recopilar más información sobre un ataque. Principalmente es una herramienta para la investigación debido a que son soluciones costosas de desplegar y mantener y no merece la pena hacer todo el esfuerzo de construir y mantener una honeynet sólo para detectar ataques.

Sin embargo, la aparición de honeynets virtuales las han hecho más asequibles para todo tipo de entornos al poder utilizar un único ordenador para alojar los señuelos y los distintos sistemas de control. [4]

2.4. IDS e IPS

Como se comentaba en la sección *2.1.1. Seguridad Perimetral*, existen diferentes herramientas para proteger el perímetro. En este apartado se va a hablar sobre los sistemas IDS e IPS y la manera en la que aumentan la seguridad de las redes.

Los sistemas de detección de intrusos (IDS) y los sistemas de protección de intrusos (IPS) no se basan únicamente en las comunicaciones basadas en direcciones IP y puertos como los firewalls, sino que además revisan el tráfico de red o el comportamiento de los equipos para detectar actividad maliciosa. La principal diferencia entre ambos sistemas es la forma de actuar al detectar un ataque.

2.4.1. IDS

El Sistema de Detección de Intrusos añade seguridad preventiva a la red ya que ante cualquier actividad sospechosa envía alertas anticipadas a los administradores de sistemas. No obstante, no está diseñado para detener ataques como el sistema IPS [12].
Existen dos tipos de IDS en función de la localización del mismo:

- HIDS (Sistema de Detección de Intrusos de Host): Este IDS se encuentra en un host y analiza la información de las actividades realizadas en él por un atacante. Tras analizar la información si encuentra modificaciones en el host, hace un reporte con sus conclusiones.
- NIDS (Sistemas de Detección de Intrusos de Red): Este IDS se instala en la red, por lo que con un único IDS es suficiente para detectar ataques en todo el segmento de red al cual están los equipos conectados.

Considerando las tareas que realizan los IDS, podemos distinguir entre IDS pasivo, que se encargan de realizar la prevención escuchando el tráfico, e IDS reactivos, que elaboran respuestas defensivas antes del ataque.

2.4.2. IPS

Un Sistema de Protección de Intrusos o IPS es un dispositivo que se encarga de controlar el acceso a una red mediante políticas de seguridad con el fin de proteger el sistema frente a posibles ataques. Si se produce un ataque, lo analiza y lo detiene antes de que se pueda ejecutar.

En función de cómo detecta el tráfico malicioso un IPS se categoriza de la manera siguiente:

- Detección basada en firmas: Detecta ataques de forma eficiente a partir de una información previa, sin embargo, es prácticamente inservible para detectar ataques aún no conocidos por el sistema.
- Detección basada en políticas: Sistemas muy efectivos, pero necesitan una configuración detallada.
- Detección basada en anomalías: Sistemas que buscan detectar comportamientos anómalos a partir de patrones.

2.5. Acceso remoto y redes virtuales

Como se comentaba en secciones anteriores, el límite del perímetro tradicional de la seguridad de las redes ha cambiado debido al traslado de aplicaciones a la nube y el uso de dispositivos.

Por ello, se utiliza el acceso remoto que permite acceder desde un dispositivo a otro y realizar tareas en él sin tener que estar físicamente en contacto, es decir, permite realizar las tareas a distancia.

Una de las formas habituales de realizar acceso remoto seguro es desplegar redes privadas virtuales (VPN).

Una VPN (*Virtual Private Network*) es una tecnología de red que permite una extensión segura de la red de área local (LAN) sobre una red pública. Esto permite conectarse directamente a una red privada y aislarse del resto del tráfico de la red pública (como, por ejemplo, el Wi-fi de un restaurante) [13].

2.5.1. Funcionamiento de una VPN

Una red privada virtual se basa en un protocolo denominado protocolo de túnel, que cifra los datos que se transmiten desde un lado del VPN al otro.

Cuando se realiza una conexión a Internet, el tráfico que le llega al proveedor de Internet está cifrado ya que el tráfico una vez pasado por el proveedor se dirige directamente al servidor VPN (servidor de acceso remoto) y se utiliza la IP de este servidor, que será el encargado de tramitar la solicitud de manera cifrada. El equipo remoto proporciona los datos al servidor VPN, y este envía la respuesta cifrada. Una vez que el cliente VPN del usuario recibe los datos, los descifra y los envía al usuario.

2.5.1.1. Despliegue de una VPN

Las dos tecnologías más utilizadas para desplegar las redes privadas virtuales son TLS e IPSec [14].

- El protocolo TLS (*Transport Layer Security*) es un protocolo criptográfico que se utiliza para transferir datos de manera cifrada a nivel de capa de transporte. Permite la autenticación del cliente y servidor usando claves públicas y certificados digitales y proporciona comunicación segura mediante el cifrado de la información entre emisor y receptor.
- El protocolo IPSec (*Internet Protocol Security*) es un protocolo que se utiliza para transferir datos de manera segura a nivel de capa de red cuya función es

mejorar la seguridad del protocolo IP para garantizar la privacidad, integridad y autenticación de los datos enviados. Existen dos modos de funcionamiento de este protocolo, modo transporte y modo túnel que es el relacionado con las redes privadas virtuales. En el modo túnel, todo el paquete IP (datos más cabeceras del mensaje) es cifrado y/o autenticado. Una vez hecho esto, se encapsula en un nuevo paquete IP y se le añade una copia de la copia de la cabecera del paquete IP original para que se realice el enrutamiento. El objetivo de este modo es establecer una comunicación segura IPSec, tras establecer una asociación de seguridad (SA), entre dos redes remotas sobre un canal inseguro.

Los protocolos TLS e IPSec son propuestas complementarias, aunque VPN TLS es más popular debido a que cubre de manera más económica y sencilla el acceso remoto.

Las principales ventajas que destacar de TLS frente a IPSec, es que no requiere desplegar agentes software de manera permanente sobre los dispositivos a los que permite acceder a aplicaciones corporativas y permite acceder a un mayor número de personas de forma remota a través de Internet siendo difícil de desplegar esta opción con IPSec.

Además, ya que trabaja a nivel de capa de transporte, permite controlar los recursos a los que un determinado usuario está autorizado a acceder. Por lo tanto, TLS conlleva a una disminución de la carga de despliegue y soporte frente a IPSec, simplifica la gestión y mantenimiento y reduce el coste de la solución.

Por otro lado, el protocolo IPSec ofrece ventajas cuando se trata de realizar conexiones LAN to LAN a este nivel, permitiendo que el usuario tenga la misma experiencia que si estuviera ubicado en la propia LAN a la que accede. Sus limitaciones en relación con los dispositivos cliente soportados puede verse como una ventaja ya que eleva la seguridad del acceso evitando potenciales ataques de seguridad en la red.

Además, permite trabajar con todo tipo de recursos y aplicaciones sin tener que instalar diferentes soluciones a diferencia de TLS VPN, que sólo permite el acceso a quien dispone de soporte Web, no permite el acceso a estaciones de trabajo, ni soporta tráfico de voz ni streaming. Por lo tanto, IPSec es la mejor propuesta para conectar múltiples redes privadas debido a su capacidad de distribuir conectividad completa a nivel de red y por limitaciones comentadas de TLS [15].

En conclusión, debido a las diferentes posibilidades y ventajas de ambas propuestas, se utilizan de forma complementaria, ocupándose IPSec VPN de las conexiones sitio a sitio y TLS VPN del acceso remoto a través de Internet.

2.5.2. Utilización de las conexiones VPN

Las conexiones VPN se utilizan para diversas utilidades entre las que destacan el teletrabajo, que permite realizar el mismo cometido a un trabajador sin estar presencialmente en su lugar de trabajo de forma remota y se garantiza que el acceso a la empresa mediante una conexión VPN está cifrada, otra funcionar más es que evita censura y bloqueos geográficos de contenido, ya que permite acceder a páginas bloqueadas en nuestro país y evitar censuras debido a que con la propia tecnología VPN el equipo se conecta al ISP que da el servicio y de ahí al VPN, obteniendo una IP diferente que no será bloqueada al ser una IP de la página del país al que nos conectemos con el fin de falsear nuestra ubicación.

Otra utilidad que destacar es el aumento de seguridad y privacidad ya que se cifran los paquetes que pasan por esta red, aumentando la seguridad del tráfico, también se ocultan datos de navegación permitiendo que se puedan realizar conexiones a redes Wi-Fi públicas, como restaurantes o cafeterías, y que la conexión sea privada sin intercepción de nuestra información e incluso realizar descargas P2P, sin embargo, cada vez hay más proveedores que las bloquean.

2.5.3. Ventajas e inconvenientes de las conexiones VPN

Las ventajas principales del uso de VPN son que permite falsear la ubicación, como se comentaba en la sección *2.5.2 Utilización de las conexiones VPN*, permite utilizarse en todas las aplicaciones, ya que enruta todo el tráfico de Internet, es fácil de utilizar, ya que la red se conecta y se desconecta fácilmente tras su configuración, facilita la comunicación entre usuarios que se encuentran en diferentes lugares, el ISP no puede saber que páginas visitas en Internet, sin embargo, la compañía que gestiona el VPN sí, por lo que debe ser de confianza y sobre todo destacar el aumento de la seguridad que proporciona en la red [16].

En cuanto a inconvenientes en el uso de una VPN destaca que existen servicios VPN gratuitos, sin embargo, no garantizan las mismas funcionalidades que servicios VPN de pago, la velocidad de conexión, subida y descarga se puede ver ralentizada en función de la localización del servidor VPN, aumenta la seguridad, pero no la garantiza

por completo, no proporciona el anonimato y no siempre consigue falsear la ubicación [16].

Capítulo 3: Plataforma escogida para soluciones de seguridad en redes de bajo coste

Una vez explicado las diferentes maneras de proteger el perímetro y aumentar la seguridad de las redes, en este apartado se va a explicar más en detalle el porqué de la elección de la plataforma Raspberry Pi (su principal funcionamiento, las características a destacar y ventajas y desventajas) y algunas pinceladas sobre otras plataformas.

3.1. Introducción a las plataformas de bajo coste

Hoy en día, existen diversas plataformas y prototipos de hardware libre orientados al mundo IoT (*Internet Of Things*) que permiten crear diferentes proyectos de una forma sencilla, están orientadas al público no experto y son fáciles de programar. Tienen el objetivo de solucionar problemas o crear proyectos caseros sin necesidad de realizar un esfuerzo económico en la compra de microprocesadores y sus respectivos componentes. El hardware y características que tienen algunas plataformas de bajo coste las capacitan para poder desarrollar proyectos que permiten garantizar la seguridad.

3.1.1. ARM

Los procesadores ARM son de tipo RISC (*Reduced Instruction Set Computer*) con 32 bits en una primera instancia ya que también existen a día de hoy de 64 bits. ARM se creó como una arquitectura para ordenadores y dispositivos de bolsillo, pero en la actualidad se encuentra en más dispositivos como grabadores de vídeo, reproductores, tabletas, teléfonos, televisores, coches, etc. Como propiedades a destacar, tienen instrucciones de tamaño fijo con pocos formatos y sólo las instrucciones de carga y almacenamiento acceden a la memoria de datos por lo tanto maneja un sistema de instrucciones simple llevando así a una utilización mínima de energía y calor. Basados en ARM han sido desarrollados procesadores como el ARM7, ARM9, ARM11 y Cortex y se encuentran en multitud de dispositivos. Por ello, el objetivo de diseñar máquinas que utilizan procesadores ARM es facilitar el paralelismo en la ejecución de instrucciones y como consecuencia permitir realizar pequeñas tareas con procesos más cortos [17].

En cuanto a software, es compatible con sistemas operativos como Linux, iOS o Android y en cuanto a Microsoft, solo con Windows 10.

3.1.2. Arduino

Una de las plataformas más conocidas es Arduino, se creó en el Instituto de Interacción Ivrea, es open-source o de código abierto basada en hardware y software flexibles y fáciles de usar, pudiéndose ensamblar las placas o comprarlas montadas. Se puede descargar el software de forma gratuita. Es una forma sencilla de realizar diferentes proyectos interactivos o crear objetos, además debido a los receptores de entrada que tiene incorporados puede interactuar con el entorno mediante sensores. Está dirigido principalmente a estudiantes, profesores o aficionados interesados en este tema.

A diferencia de otras plataformas o microcontroladores, los entornos de desarrollo y el lenguaje de programación de Arduino han sido creados paralelamente, por lo cual no existe ningún problema de compatibilidad.

En la parte de software, el microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Existe un IDE (*Integrated Development Environment*) para casi todas las plataformas (Windows, Linux, Mac).

En la parte de hardware, existen muchas placas basadas en Arduino debido a que es un hardware open-source, por lo que existe Arduino de todos los tamaños, colores y funcionalidades. El hardware Arduino más sencillo consiste en una placa con un microcontrolador y una serie de puertos de entrada y salida.

3.1.2.1. ¿Para qué sirve Arduino?

Arduino se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado a otros dispositivos hardware o interactuar con otros softwares.

Por lo tanto, el uso de Arduino se puede dividir en Arduino utilizado como microcontrolador, que controla dispositivos y toma decisiones a partir de un programa interactuando así con el mundo físico mediante sensores y actuadores, y como interfaz entre un ordenador y otro dispositivo, convirtiendo una tarea en una acción en el mundo físico y, al contrario, que el ordenador ejecute una acción a partir de los sensores conectados a Arduino [18].

Las placas Arduino cuentan con Shields, que son placas de expansión con las que pueden aumentar su capacidad y pueden realizar infinidad de funciones como GPS,

relojes en tiempo real, conectividad por radio, pantallas táctiles LCD, placas de desarrollo, y un larguísimo etcétera de elementos.

Orientado al tema de la seguridad, es una plataforma que mediante la programación de código y de la propia placa permite monitorizar el tráfico o realizar tareas de backups, además gracias a los sensores que conectan la placa con el mundo, se puede implementar que si ocurre algún ataque informático a una plataforma u ordenador asociado informe de forma física mediante bombillas o leds.

3.1.3. Raspberry Pi

Raspberry Pi es una plataforma u ordenador de bajo coste creada por la fundación Raspberry Pi con ayuda de la universidad de Cambridge, con el objetivo de facilitar el aprendizaje sobre aspectos de la informática como aprender a programar o crear pequeños proyectos hardware a nivel educativo. Por ello, la Raspberry Pi fue creada con el fin de llegar al mayor número de usuarios y costar lo menos posible.

En la parte de software, existen diferentes sistemas operativos que se instalan sobre una tarjeta SD / micro SD ya que la plataforma cuenta con un lector de tarjetas SD. Soporta Linux con diversas distribuciones, siendo Raspbian la más conocida, FreeBSD, Android y Windows 10. Otra opción, es instalar un gestor de contenido multimedia (como OSMC [19], OpenElec [20]) y utilizar así la Raspberry Pi como un media center. En cuanto a lenguajes de programación, Python es el lenguaje de programación principal de Raspberry Pi, aunque pueden utilizarse otros como C, C#, Ruby, Java, Perl y BASIC.

En la parte de hardware [21] [22], tiene un tamaño reducido, una alta capacidad de procesamiento en función de su tamaño y un coste bajo lo que le hace ser una placa ideal para utilidades de seguridad Informática y redes. El microprocesador ha ido mejorando y añadiendo nuevos elementos con la evolución de los modelos. Los primeros modelos que salieron al mercado fueron Raspberry Pi 1 Model A y Raspberry Pi 1 Model B y B+.

La Raspberry Pi A usa un procesador gráfico VideoCore IV de 250 MHz, un chip Broadcom BCM2835 con un CPU ARM1176JZF a 700 MHz, tiene 256 MB de RAM, un puerto USB, conexión HDMI, salida Jack de 3.5 mm y no dispone de puerto Ethernet. Esta placa ha sido sustituida por la Raspberry Pi Model A+, una versión mejorada y al mismo precio que tiene más conectores GPIO (General Purpose

Input/Output), consume menos energía, menor tamaño, audio mejorado y soporte Micro- SD.

La Raspberry Pi B cambia respecto de la Raspberry Pi Model A en que tiene 512 MB de memoria RAM, 2 puertos USB y un puerto Ethernet y la Raspberry Pi Model B+ en que tiene 4 puertos USB y la salida SD ha sido sustituida por una microSD, el resto del hardware de ambas placas no varía respecto de la Raspberry Pi Model A.

En el año 2014, se lanzó el modelo Raspberry Pi 2 Model B, con un chip BCM2836 con más MHz, pero con la misma arquitectura, se mejora la memoria RAM con 1 GB, una nueva CPU ARM Cortex-A7 de cuatro núcleos a 900 MHz que es 6 veces más potente que el de los anteriores modelos además de ser compatible con Windows 10 y Ubuntu.

La Raspberry Pi Zero es un modelo reducido en tamaño y en precio, tiene dos puertos micro-USB, una ranura Micro-SD y una salida de vídeo mini-HDMI, pero no tiene puertos de conexión (USB, HDMI, Ethernet). Sin embargo, es un 50% más potente que la Raspberry Pi Model B.

La Raspberry Pi 3 Model B, del año 2016, incluye un nuevo procesador ARM Cortex A53, mantiene cuatro núcleos respecto a la Raspberry Pi 2 Model B, pero de 1.2 GHz de 64 bits siendo así 10 veces mejor que la Raspberry Pi original e incluso un 50% mejor que la Raspberry Pi 2. Destacar de este modelo la adición de Bluetooth y Wi-fi sin necesidad de adaptadores. Y el último modelo, la Raspberry Pi 3 B+, Marzo del 2018, actualiza el modelo anterior, cuenta con un nuevo procesador de 1.4 GHz y mejor conectividad, en cuanto a la conectividad inalámbrica incorpora doble banda a 2,4GHz y 5GHz, y su nuevo puerto Ethernet de 300 Mbits/s, también cuenta con Bluetooth 4.2 y Bluetooth BLE.

3.1.3.1. Principales características y ventajas de Raspberry Pi

Las principales características y ventajas que destacar de esta plataforma son parecidas a las características de la placa Arduino, ya que tienen bastantes cosas en común:

- Asequible. Los modelos Raspberry Pi son baratos, no superando los 40€ las placas más actuales, ya que como se comentaba anteriormente tienen en principio un fin educativo siendo capaces de realizar tareas simples y complejas.
- Disponible. La adquisición de modelos Raspberry Pi, por un bajo precio, como se comentaba en la característica anterior, es muy fácil debido a la venta de los

mismos en la página oficial [34] de Raspberry Pi además de en varios portales de compra/venta en Internet e incluso en tiendas físicas.

- Multiplataforma. Permite la utilización de diferentes sistemas operativos como Linux (un gran número de distribuciones), Android y Windows 10.
- Conocida. La ventaja principal de este pequeño ordenador frente a otras plataformas es la existencia de una gran cantidad de proyectos y desarrollos de la propia comunidad de Raspberry Pi.

3.1.3.2. ¿Para qué sirve Raspberry Pi?

Esta plataforma permite realizar desde proyectos sencillos como una estación meteorológica o un media center hasta proyectos más complejos como una máquina de arcade, por ello hay que saber un mínimo de programación o computación.

Otra característica es que permite realizar proyectos son los conectores GPIO (General Purpose Input/Output) que representan la interfaz entre la Raspberry Pi y el mundo exterior. Son un número de pines que se configuran como entradas o salidas digitales para controlar sensores, periféricos o actuadores externos como encender y apagar LEDs, obtener el estado de un interruptor, etc [23].

También permite conectar Arduino y la Raspberry Pi mediante estos pines consiguiendo así grandes proyectos utilizando ambas plataformas.

Relacionado con los aspectos de seguridad que es el tema que compete, debido a sus características hardware y software y su potencia como placa, permite realizar proyectos relacionados con la seguridad, además existen software de código abierto creados especialmente para la Raspberry Pi que permiten controlar el acceso a los datos de un sistema, definir acciones defensivas frente a amenazas o ataques, monitorizar la red, etc. Un ejemplo es la herramienta Fail2ban [24] que permite asegurar la placa frente ataques de fuerza bruta escaneando automáticamente los logs.

Como conclusión de esta plataforma, destaca por su potencia de cálculo, memoria y frecuencia del procesador a comparación con Arduino. Puede sustituir a un ordenador en tareas simples y existe numerosa documentación sobre esta placa a nivel técnico, ejemplos de iniciación, proyectos más complejos, etc. Por lo tanto, esta placa se puede utilizar como una herramienta de seguridad.

3.1.4. Otras plataformas

Aunque las placas de desarrollo Arduino y Raspberry Pi son de las más famosas, existen otras placas en el mercado que pueden ser posibles competidoras [25].

- Una placa interesante es Gooseberry, tiene un procesador A10 ARM Cortex-A8, gráficos Mali 400, memoria RAM de 512 MB, 4 GB de almacenamiento interno, Wi-fi, entrada para tarjeta microSD, puerto mini HDMI, USB y viene precargado con sistema operativo Android. Es un avance respecto a la Raspberry Pi teniendo más velocidad y siendo más cara, pero se hicieron muy pocas unidades.
- La placa Hackberry tiene características similares a la placa Cubieboard, la diferencia es que tiene un procesador a 1.2 GHz y no cuenta con interfaz SATA mejorando así las características de la Raspberry Pi.
- Otra placa de hardware libre es BeagleBone, está entre medio de las placas Arduino y Raspberry Pi, ya que tiene potencia para ejecutar sistemas operativos, pero mantiene la sencillez y flexibilidad de Arduino. Tiene un procesador ARM Cortex-A8 que corre a 720 GHz, 256 MB de memoria RAM, puerto Ethernet y USB y entrada para tarjeta microSD entre sus principales características.
- Orange Pi es un clon de código abierto de la Raspberry Pi, tiene un procesador ARM Cortex-7 de cuatro núcleos con un 1 GB de memoria RAM además de 8 GB de memoria flash. Se puede ejecutar Android, Ubuntu, Debían, así como las imágenes de Raspberry Pi y Banana Pi (otra placa de código abierto). Esta placa la puede utilizar cualquier persona que quiera empezar a crear tecnología, al igual que el resto de las placas comentadas.

3.2. Elección de la plataforma de bajo coste

Los principales criterios para elegir una placa u otra han sido principalmente el coste de la misma, la facilidad de uso ya que se busca una plataforma que sea sencilla, la disponibilidad de proyectos de seguridad abiertos para tomarlos como referencia además de tener una comunidad de apoyo que permita solventar dudas. Por ello, la elección de la plataforma se ha debatido entre las placas Raspberry Pi y Arduino ya que ambas son buenas plataformas para la creación de. Hoy en día existen plataformas con mejores características que estas dos placas a nivel de hardware, pero el éxito de las dos se debe a que son baratas, versátiles y existe información prácticamente ilimitada sobre ellas en Internet.

El condicionante para la elección de la plataforma se debe a las principales diferencias [26] que tienen ambas placas. Arduino es un microcontrolador, mientras que Raspberry Pi es un microprocesador, siendo el segundo un miniordenador completo y se puede así ejecutar un sistema operativo completo. Otra diferencia es que Raspberry Pi cuenta con una salida Ethernet, varias salidas USB y HDMI mientras que Arduino necesita una caja de expansión (shield) para tener una salida Ethernet.

En cuanto a proyectos, Arduino se utiliza más para proyectos de electrónica o robótica ya que permite conectar multitud de sensores y actuadores, por otro lado, la Raspberry Pi gracias a los pines GPIO también permite conectar elementos electrónicos e interconectar nuestro proyecto con el mundo como lo haríamos con un Arduino, aunque es más complicada y menos directa para proyectos sencillos, por ello se utiliza más para proyectos más complejos y de programación.

Referido a características técnicas Raspberry Pi cuenta con una salida HDMI lo que permite conectar un monitor HD para obtener imágenes en alta definición, tiene una velocidad mayor de placa, ya que cuenta con 700 MHz (aunque el procesador puede ser forzado a más velocidad), mientras que en Arduino la velocidad es de 16 MHz, y puede sustituir a un ordenador en tareas simples además de poder usar sus puertos de conexión para interconectar nuestro proyecto con el mundo como lo haríamos con un Arduino.

Por ello, como conclusión de estas diferencias es que, si se busca hacer un proyecto más electrónico o relacionado con la robótica, la opción a elegir es la placa Arduino ya que permite conectar sensores y componentes de manera sencilla y rápida. Por otro lado, la Raspberry Pi, como se comentaba en uno de los puntos anteriores, cuenta con pines GPIO que también permite conectar sensores, pero está pensada para otros proyectos.

3.2.1. ¿Qué se necesita?

Tras la elección de la placa Raspberry Pi para realizar este proyecto, en esta sección se va a explicar los materiales que se necesitan para poner en funcionamiento la placa.

- Raspberry pi: Existen diversos modelos de Raspberry Pi como se comentaba en la sección 3.1.3 Raspberry Pi, para este proyecto se ha elegido la placa Raspberry Pi 2 Model B. Es seis veces más potente que las versiones anteriores y tiene más memoria hardware que el modelo B y B+. Aunque la placa más reciente sea la Raspberry Pi 3 Model B, la Raspberry Pi 2 Model B es más

económica y tiene las características necesarias para la realización de este proyecto. El precio de esta placa, en mi caso, ha sido de 26.07€

- Monitor o Televisión: Se necesita un monitor para poder visualizar el sistema operativo y poder trabajar con él. Es importante que tenga conector VGA para poder utilizar un cable HDMI-VGA, si no es así, se puede utilizar un adaptador HDMI – DVI o la salida de vídeo compuesto y de audio a través de un minijack. Ya se contaba con un monitor por lo que el precio ha sido de 0€.
- Cable HDMI: Para conectar la Raspberry Pi con el monitor en el caso que tenga conector VGA se necesita un cable HDMI, segundo material de la Figura 3.4. Si el monitor tiene conector DVI, será necesario comprar un adaptador HDMI – DVI. El precio del cable HDMI 2.0 de 1.8 metros ha sido de 7,29€. La Raspberry Pi se puede conectar mediante escritorio remoto tras haberlo configurado en la misma.
- Teclado USB: Se necesita un teclado para poder introducir el texto en la Raspberry Pi. Ya se contaba con este material, por lo que el precio ha sido de 0€.
- Ratón USB: Para poder controlar el cursor o puntero del ratón en el ordenador o en la Raspberry Pi. Ya se contaba con este material, por lo que el precio ha sido de 0€.
- Fuente de alimentación: Se recomienda una fuente de alimentación europea, tercer material de la Figura 3.4, de 5V, 2.5A para conectar la Raspberry Pi. El precio de la fuente de alimentación de 5 V ha sido de 9,99€.
- Tarjeta SD: En la tarjeta SD, cuarto material de la Figura 3.4, estará el sistema operativo a utilizar, por ello, se recomienda que como mínimo sea de 4 GB. Se ha comprado una tarjeta SD de 32 GB por un precio de 5,12€.

Estos son los accesorios indispensables para poder iniciar un proyecto con la plataforma Raspberry Pi. Además, se ha comprado una caja y unos disipadores, quinto material de la Figura 3.4, para almacenar la Raspberry Pi y protegerla del calor por 6,90€. En el caso que se quiera conectar la Raspberry Pi a Internet, se necesitará un cable Ethernet conectándose un extremo en la misma y el otro extremo en un punto de acceso a la red, como por ejemplo un router.

En total se puede obtener una plataforma muy potente por un precio muy económico en torno a los 50€ con solo los accesorios indispensables.

3.2.2. ¿Qué sistema operativo utilizar?

El siguiente paso tras tener todos los elementos necesarios es elegir un sistema operativo para instalar en la tarjeta SD. Debido a que utiliza una arquitectura ARM y no x86-64, no es compatible con todos los sistemas operativos, pero existen multitud de ellos que si lo son y se utilizan en función del proyecto que se quiera realizar y el uso que se quiera dar al hardware. La mayoría de los sistemas operativos son distribuciones Linux, existiendo así distribuciones oficiales y no oficiales.

Existe la aplicación NOBS (New Out Of Box Software), que, aunque ocupe más espacio en la tarjeta SD o microSD, permite la instalación de diversas distribuciones Linux que se descargan de Internet además de venir con el sistema operativo Raspbian precargado. Se puede comprar una tarjeta SD con el software ya instalado o descargar el software de la página oficial [40] y arrastrar los archivos a una tarjeta SD con un mínimo de 4 GB una vez formateada. Para cambiar de SO solo se tiene que pulsar la tecla Shift en el teclado al iniciar la Raspberry Pi.

El sistema operativo oficial más recomendable para iniciarse con la Raspberry Pi es Raspbian. Raspbian es una distribución del sistema operativo GNU/Linux basada en Debian, pero optimizada para el hardware de la Raspberry Pi. Puede instalarse directamente en la tarjeta SD que se conectará a la Raspberry Pi descargando el software de la página oficial, creando una tarjeta SD booteable “quemando” la imagen con cualquier programa que realice esta tarea como Rufus [27] o Etcher [28].

Capítulo 4: Implementación de un servidor VPN en una plataforma Raspberry Pi

En este capítulo se va a desarrollar de una manera didáctica la solución elegida para securizar las redes a las que se conectan los usuarios con una plataforma de bajo coste y un consumo mínimo de energía.

En primer lugar, se va a explicar que es un servidor VPN y cuál es su finalidad. A continuación, se va a desarrollar su implementación y las configuraciones que se han de tener en cuenta.

4.1. VPN

El objetivo de una VPN (Virtual Private Network), como se comenta en la sección 1.5. Acceso remoto y redes virtuales, es extender la propia red privada del

usuario a redes públicas como cafeterías, centros comerciales, etc. con el objetivo de que la navegación del usuario por Internet permanezca cifrada.

Existen diferentes maneras de configurar una VPN, se ofrecen servicios gratuitos y servicios de pago. Los servicios gratuitos no son para nada recomendables ya que la mayoría de las empresas/compañías que ofrecen este servicio gratuito recopilan los datos de los usuarios con el fin de venderlos e incluso realizar Man in the Middle para sustituir los datos por los que quiera el atacante. Además, la velocidad o uso de este servicio suele estar limitado, suele ser de mala calidad e incluso la empresa puede dejar de ofrecer el servicio sin avisar por diversos motivos como legales o económicos. En cuanto a los servicios de pago, hay que tener en cuenta el servicio que ofrece la empresa, su trayectoria y las opiniones de los clientes. Sin embargo, existe la opción de crear una VPN privada que es gratuita y se puede configurar como el usuario quiera, lo único necesario es el hardware que se comentaba en el *Capítulo 2*, y alguna noción de informática.

4.1.1. Protocolos para la implementación del servidor VPN

Existen diferentes protocolos que se pueden utilizar para montar una VPN, algunos de ellos son PPTP (Point to Point Tunnelling Protocol) aunque ya no se recomienda su utilización debido a que han sido encontradas vulnerabilidades, IPSec/L2TP (Layer 2 Tunneling Protocol) donde L2TP se encarga de establecer el túnel entre el origen y el destino utilizando el protocolo UDP e IPSec se encarga de garantizar las comunicaciones en el nivel IP de la pila de protocolos TPC/IP, SSTP (Secure Software Tunneling Protocol) que permite encapsular tráfico PPP(Point-to-Point Protocol) sobre una conexión SSL, IKEv2 es un túnel VPN basado en IPSec considerado incluso mejor que IPSec/L2TP en cuestiones de seguridad, rendimiento y estabilidad y por último OpenVPN que es una solución de software libre con licencia GNU que es capaz de crear VPNs utilizando SSL/TLS para el intercambio de claves además de utilizar OpenSSL como librería criptográfica [29] .

Finalmente, se ha elegido el protocolo OpenVPN para la implementación de la VPN ya que es un protocolo que cuenta con mecanismos de cifrado sin sufrir la complejidad de otras soluciones VPN como IPSec de intercambio de claves, tiene una alta flexibilidad y posibilidad de extensión mediante scripting, tiene soporte para proxy, las conexiones OpenVPN pueden ser realizadas a través de casi cualquier firewall, tiene protección de los usuarios remotos ya que una vez que OpenVPN ha establecido un

túnel el firewall de la organización protegerá el laptop remoto aun cuando no es un equipo de la red local además de no ser muy complicado de implementar en la Raspberry Pi en este caso [30] .

4.2. Desarrollo de la solución elegida

Los requisitos previos para comenzar a desarrollar el servidor VPN son los siguientes:

- Tener instalado un sistema operativo en la Raspberry Pi, en este caso Raspbian como se explica en el anexo A. Configuración y puesta en marcha de la Raspberry Pi.
- Tener una IP estática para la Raspberry Pi en la red doméstica, como se explica en la sección A.2. Conexión vía SSH a la Raspberry Pi.
- Habilitar el SSH, pudiendo prescindir así de un monitor y teclado y poder acceder de forma remota desde cualquier ordenador, como se explica en el anexo A.1.Configuración de aspectos iniciales.
- Reenviar el tráfico al puerto 1194, que es el que se utiliza para el servicio VPN. Esta configuración se explica a continuación en la sección *4.2.1.1. Configuración e instalación de OpenVPN*.

Una vez se cumplan estos requisitos, se está preparado para comenzar a crear el servidor VPN como se explica a continuación de la configuración e instalación de OpenVPN con ayuda de código y figuras.

4.2.1. Lado del servidor

Se van a diferenciar dos partes a la hora de explicar de forma didáctica la implementación del servidor VPN. En esta sección se va a explicar la parte del servidor que es la parte que se ocupa de la seguridad y los datos sensibles.

4.2.1.1. Configuración e instalación de OpenVPN

Como se explica en el Anexo *A.2.Conexión vía SSH a la Raspberry Pi*, se puede establecer una IP estática para poder acceder desde SSH y así si se reinicia la placa no se modifica la IP de la misma.

A continuación, se debe realizar una serie de configuraciones en el router para que el servicio VPN funcione correctamente.

Para que el tráfico pase por la Raspberry Pi y funcione el servicio VPN, se debe redireccionar ese tráfico al puerto 1194 que es el puerto predeterminado para configurar

un servidor VPN. La forma de redireccionar el tráfico depende del router, pero suele realizarse declarando una nueva regla en la pestaña Port Forwarding. En mi caso, cuento con un router ZTE- y se haría de este modo:

En primer lugar, se accede a la configuración del router poniendo en el navegador la dirección 192.168.1.1, una vez dentro se muestra una pantalla de login que en función del modelo del router es un usuario y contraseña predeterminado. En este caso, es nombre de usuario "admin" y contraseña "admin".

Una vez se esté en el panel de control de configuración del router, se accede a la pestaña Port Forwarding y se rellenan los campos con los datos adecuados como se muestra en la *Figura 4.1*.

Tras haber realizado estas configuraciones iniciales, se va a instalar OpenVPN. Para ello en primer lugar hay que actualizar el sistema operativo en busca de las últimas versiones:

```
sudo apt-get update
sudo apt-get upgrade
```

Una vez actualizado el sistema operativo, se realiza la instalación del paquete OpenVPN:

```
sudo apt-get install openvpn
```

Debido a que la instalación ocupa espacio, se pregunta si se está seguro de la instalación, pulsar "y" para aceptar instalar las dependencias.

4.2.1.2. Generación de claves

Para proteger el servidor y que nadie pueda ver los paquetes, se van a generar claves RSA públicas y privadas. El sistema criptográfico RSA es un algoritmo asimétrico que utiliza una clave pública para cifrar el contenido, que es la que se distribuye, y una clave privada que es la que se queda el propietario para descifrar el mensaje. Es el sistema más conocido y usado en los sistemas de clave pública.

OpenVPN viene con un paquete llamado easy_rsa que permite utilizar este método de encriptación de una manera sencilla para generar las claves.

Para generar las claves se necesita tener permisos de superusuario, para eso se ejecuta el comando sudo cambiando así la línea de comandos de pi@raspberrypi a root@raspberrypi.

A continuación, se va a copiar de forma recursiva el contenido de la carpeta easy- rsa creada al instalar openvpn en otra carpeta con el mismo nombre, pero en otro directorio ya que se va a modificar algún archivo y no tocar los originales.

```
cp -r /usr/share/easy-rsa /etc/openvpn/easy-rsa
```

Una vez copiado los archivos, se cambia a ese directorio que es donde se van a encontrar las claves.

```
cd /etc/openvpn/easy-rsa
```

Se va a proceder a modificar el archivo vars que se encuentra en la carpeta actual con el fin de modificar algunas líneas y escribir la ubicación donde se generarán las claves RSA. Para ello, existe el comando nano o vi.

```
nano vars
```

Se ha de buscar la línea en la que pone export EASY_RSA= “” y modificarla por export EASY_RSA = “/etc/openvpn/easy-rsa” como se muestra en la *Figura 4.2*.

La línea export KEY_SIZE debe tener valor 1024, que es el método de cifrado, o se puede cambiar por 2048 aumentando así la seguridad del cifrado, pero también hace que la generación de las claves tarde más tiempo. Por ello en este caso se ha dejado la línea como se muestra en la *Figura 4.3.*

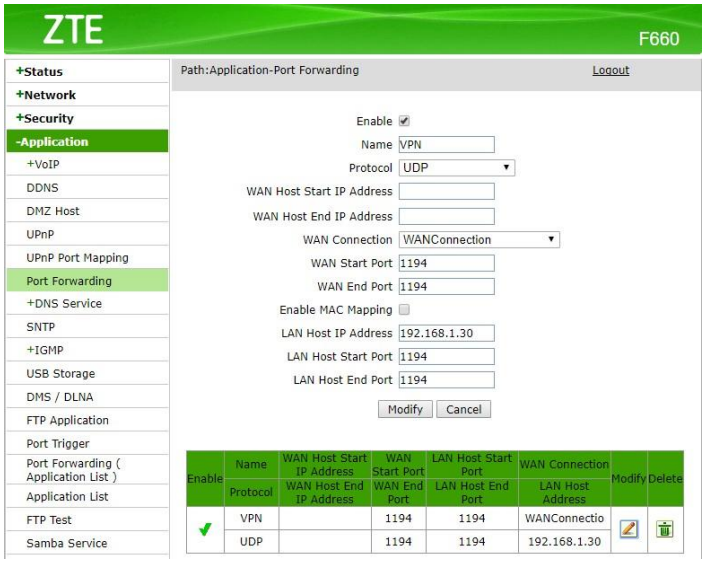


Figura 4.1. Configuración del tráfico en un router ZTE.

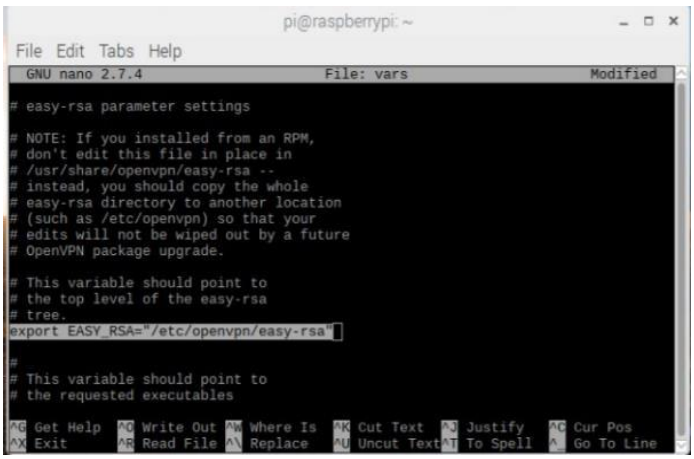


Figura 4.2. Modificación del fichero vars.

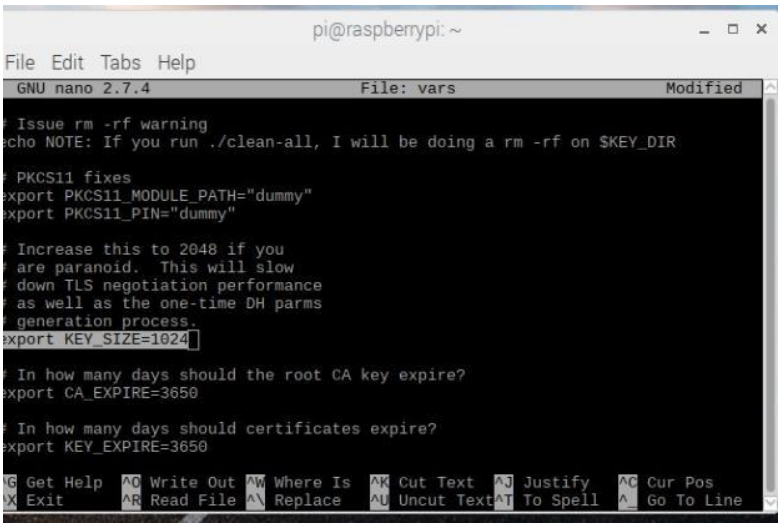


Figura 4.3. Modificación del fichero vars.

Una vez modificado el archivo, para salir y guardar los cambios hay que pulsar Control + X en el teclado.

4.2.1.3. Generación de certificados

Lo siguiente que hay que hacer es generar el certificado CA y el certificado root. Una autoridad de certificación (CA) es una entidad que emite certificados digitales, y estos certifican la propiedad de una clave pública. En este caso, se va a generar la propia entidad certificadora y firmar en las claves de OpenVPN.

Se ha de estar en el directorio /etc/openvpn/easy-rsa, y ejecutar los siguientes comandos:

`source ./vars` Este comando carga el archivo vars que se editó anteriormente.
`./clean-all` Este comando elimina las posibles claves que pudiese haber anteriormente, si no se desea eliminar las claves anteriores, no ejecutar este comando.
`./build-ca` Este comando genera la entidad certificadora.

Después de ejecutar el tercer comando, se muestran unos campos opcionales que pueden ser rellenados o no (*Figura 4.4*). Tras crear los certificados, se debe dar un nombre al servidor para crear su certificado. En este caso, se ha llamado “piserver”.

`./bulid-key-server [nombre_servidor]`

De nuevo, se vuelven a mostrar los campos opcionales que como antes pueden ser rellenados o no. Hay que tener en cuenta los tres campos siguientes:

- Nombre común: Debe ser el nombre que se puso al servidor. En este caso “piserver”.
- Contraseña de reto: Se debe dejar en blanco.
- Firmar el certificado: Se debe escribir “y”.

Y finalmente, confirmar todo “y”, el certificado será certificado por 3650 días. Con estos pasos ya se ha configurado el certificado del servidor.

A continuación, se van a crear claves para cada usuario o dispositivo. Se debe dar un nombre al usuario. En este caso, client1.

```
./build-key-pass  [username]
```

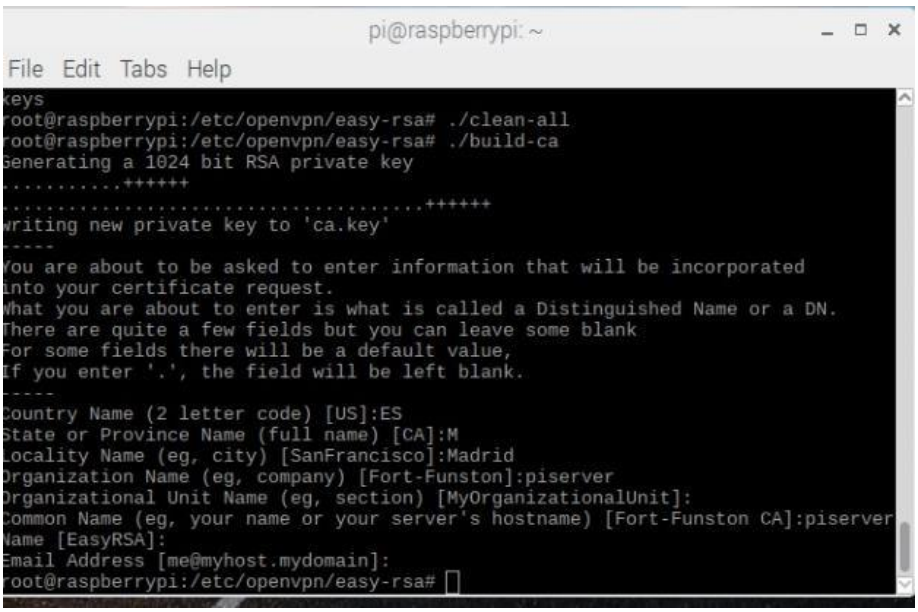


Figura 4.4. Configuración del certificado.

Una vez ejecutado el comando, vuelven a mostrarse los campos opcionales que se muestran cuando se generan las claves del servidor. En este caso, también hay que tener en cuenta los tres campos siguientes:

- Contraseña: Se debe ingresar dos veces una contraseña PEM (formato de archivo empleado para almacenar certificados digitales) que se ha de recordar.
- Contraseña de reto: Se debe dejar en blanco.
- Firmar el certificado: Se debe escribir “y”.

Tras esto, se ha de cambiar al directorio keys, y en él se va a realizar el cifrado del usuario utilizando el algoritmo de encriptación des3. Openssl es una implementación de código abierto de Secure Socket Layer, un método estándar para establecer una conexión segura. Se ha utilizado el cifrado des3 para evitar ataques de fuerza bruta. Se debe realizar este paso para cada cliente que se configure y establecer una contraseña por cada uno.

```
cd keys
openssl  rsa  -in  client1.key  -des3  -out  client1.3des.key
```

Una vez creado un certificado de servidor y en este caso, un certificado de cliente, se vuelve al directorio anterior y se genera el intercambio de claves Diffie-Hellman, el cual permite que dos entidades, sin conocimiento previo una de la otra, intercambien las claves a través de un servidor público.

```
cd /etc/openvpn/easy-rsa/
./build-dh
```

Este proceso durará unos minutos para la generación de 1024 bits, ya que utiliza números aleatorios y busca determinadas relaciones, en el caso de que se haya utilizado un cifrado de 2048 bits tardará más.

Debido a que se pueden dar ataques de denegación de servicios (DoS), se va a proteger al servidor VPN. OpenVPN evita este tipo de ataques generando la clave estática HMAC que es un código de autenticación de mensajes en clave-hash y con ello comprueba las solicitudes que se hacen al servidor, si no detecta la clave-hash no autentica la solicitud. El comando para ello es el siguiente:

```
openvpn --genkey --secret keys/ta.key
```

4.2.1.4. Configuración del servidor

Tras haber generado las claves y un CA para firmar estas claves, se va a configurar el servidor para indicar que claves, puerto, dirección IP o conexión debe utilizar. Para ello se va a crear un archivo de configuración con el siguiente comando en el directorio especificado:

```
nano /etc/openvpn/server.conf
```

El fichero como ha sido creado está en blanco, y debe rellenarse con las siguientes líneas:

```
local 192.168.1.30 # SWAP THIS NUMBER WITH YOUR RASPBERRY PI
IP ADDRESS
dev tun
proto udp #Some people prefer to use tcp. Don't change it
if # you don't know.
port 1194
ca /etc/openvpn/easy-rsa/keys/ca.crt
cert /etc/openvpn/easy-rsa/keys/piserver.crt # SWAP WITH YOUR CRT
NAME
key /etc/openvpn/easy-rsa/keys/piserver.key # SWAP WITH YOUR #KEY
NAME
dh /etc/openvpn/easy-rsa/keys/dh1024.pem # If you changed to
#2048, change that here!
server 10.8.0.0 255.255.255.0
# server and remote endpoints
ifconfig 10.8.0.1 10.8.0.2
# Add route to Client routing table for the OpenVPN Server
push "route 10.8.0.1 255.255.255.255"
# Add route to Client routing table for the OpenVPN Subnet
push "route 10.8.0.0 255.255.255.0"
```

```
# your local subnet
push "route 192.168.1.30 255.255.255.0" # SWAP THE IP NUMBER
WITH YOUR RASPBERRY PI IP ADDRESS
# Set primary domain name server address to the SOHO Router
# If your router does not do DNS, you can use Google DNS
8.8.8.8
push "dhcp-option DNS 8.8.8.8" # This should already match
your router address and not need to be changed.
# Override the Client default gateway by using 0.0.0.0/1 and
# 128.0.0.0/1 rather than 0.0.0.0/0. This has the benefit of
overriding but not wiping out the original #default gateway.
push "redirect-gateway def1"
client-to-client
duplicate-cn
keepalive 10 120
tls-auth /etc/openvpn/easy-rsa/keys/ta.key 0
cipher AES-128-CBC
comp-lzo
user nobody
group nogroup
persist-key
persist-tun
status /var/log/openvpn-status.log 20
log /var/log/openvpn.log
verb 1
```

Las líneas marcadas en negrita deben de ser modificadas por la IP estática de la Raspberry Pi y por el nombre dado al servidor.

Debido a que la Raspberry Pi por defecto no redirige el tráfico de Internet, hay que editar el fichero de configuración sysct.

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

Hay que descomentar la línea “Uncomment the next line to enable packet forwarding for IPv4” como se muestra en la *Figura 4.5*, esto permite que se reciban y envíen paquetes. Para que se guarden los cambios del fichero pulsar Control + X. Para aplicar los cambios se ejecuta el comando:

```
sysctl –p
```

El comando sysctl configura los parámetros del kernel en tiempo de ejecución, mientras que el modificador –p indica que se cargue el fichero con los cambios que se acaban de realizar.

Una vez se tiene el servidor el funcionamiento y con acceso a Internet, hay que crear una regla de firewall ya que Raspberry Pi cuenta con un firewall integrado que bloquea las conexiones entrantes provenientes de Internet o fuentes desconocidas. Para ello se va a crear un fichero con un script que se va a ejecutar cada vez que se reinicie la placa ya que cada vez que se reinicia, el firewall se restablece a la configuración predeterminada.

```
nano /etc/firewall-openvpn-rules.sh
```

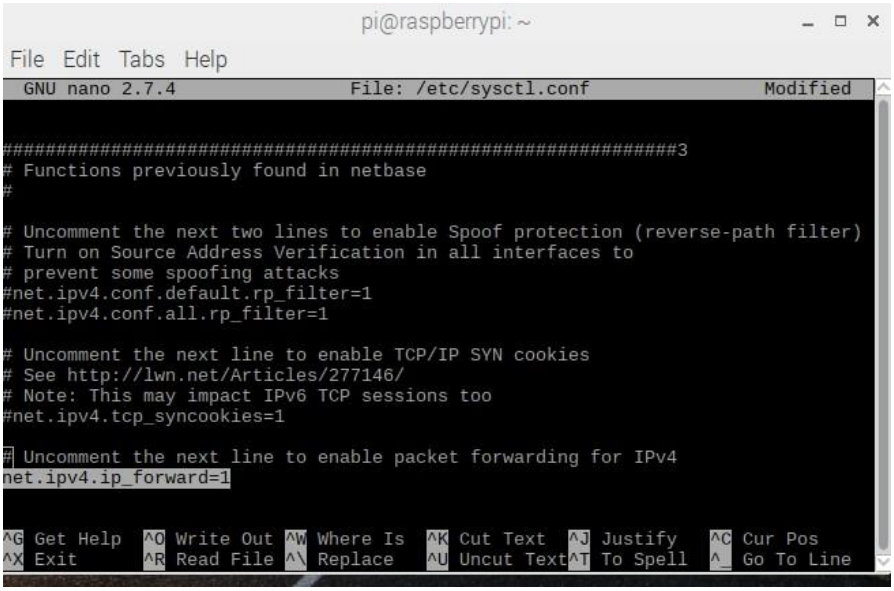


Figura 4.5. Configuración fichero sysctl.conf.

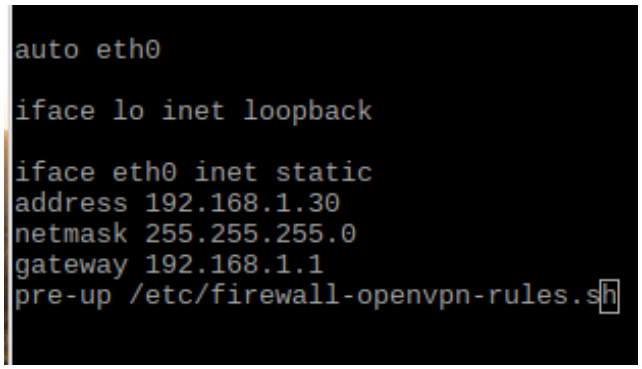


Figura 4.6. Configuración fichero interfaces.

Hay que rellenar el fichero en blanco con las siguientes líneas de script:

```
#!/bin/sh
iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o eth0 -j SNAT -- to-source 192.168.1.30
```

La IP debe ser modificada por la IP estática de la Raspberry Pi y eth0 se corresponde con nuestra tarjeta de red. En el caso de estar conectado a través de red inalámbrica la tarjeta de red debe ser modificada por wlanX. Pulse Control + X para guardar los cambios.

Se van a dar privilegios al propietario de lectura, escritura y ejecución del fichero firewall-openvpn-rules.sh creado ya que por defecto no los tienen:

```
chmod 700 /etc/firewall-openvpn-rules.sh
```

Y con este comando se cambia el propietario del fichero, en este caso al usuario root, que es el superusuario.

```
chown root /etc/firewall-openvpn-rules.sh
```

Ahora igual que antes, hay que editar el fichero interfaces para que se ejecute el script cada vez que se inicie la placa.

```
nano /etc/network/interfaces
```

Y se añaden esta línea en el fichero como se muestra en la *Figura 4.6*:

```
pre-up /etc/firewall-openvpn-rules.sh
```

Finalmente, guardar los cambios realizados en el fichero pulsando Control + X y reiniciar el sistema.

```
sudo reboot
```

Hasta aquí se ha terminado con la configuración del lado del servidor, el siguiente paso será la configuración del lado del cliente.

4.2.2. Lado del cliente

Ya se tiene un servidor VPN funcional, pero hay que configurar el lado del cliente con los usuarios creados anteriormente para que el servidor los reconozca y los asocio con las claves. Para generar un fichero de configuración para cada cliente se va a utilizar el script creado por Eric Jodoin [31].

Para empezar, hay que crear un fichero desde el cual se leerán las configuraciones por defecto que es al que tendrá acceso el script:

```
nano /etc/openvpn/easy-rsa/keys/Default.txt
```

Este fichero en blanco hay que rellenarlo con el siguiente código:

```
Client
dev tun
proto udp
remote <YOUR PUBLIC IP ADDRESS HERE> 1194
resolv-retry infinite
nobind
persist-key
persist-tun
mute-replay-warnings
ns-cert-type server
key-direction 1
cipher AES-128-CBC
comp-lzo
verb 1
mute 20
```

Se debe rellenar la línea “remote <YOUR PUBLIC IP ADDRESS HERE>” 1194 con la IP pública estática, si no se tiene se puede utilizar un servicio de sistemas de nombres de dominio dinámico (DNS) para configurar un nombre de dominio en lugar de la IP pública. En este caso se ha utilizado el servicio gratuito No-IP [32]. En el anexo, se explica cómo obtener este dominio. Una vez modificado el fichero, pulsar Control + X para guardar y salir del fichero.

A continuación, se va a crear otro fichero en el cual se pegará el script anteriormente comentado realizado por Eric Jodoin [31]:

```
nano /etc/openvpn/easy-rsa/keys/MakeOVPN.sh
```

Y cuyo código es el siguiente, que ha sido adaptado con comentarios para poder ser copiado y pegado directamente:

```
#!/bin/bash
# Default Variable Declarations
DEFAULT="Default.txt"
FILEEXT=".ovpn"
CRT=".crt"
KEY=".3des.key" CA="ca.crt"
TA="ta.key"
#Ask for a Client name
echo "Please enter an existing Client Name:"
read NAME
#1st Verify that client’s Public Key Exists
if [ ! -f $NAME$CRT ]; then
echo "[ERROR]: Client Public Key Certificate not found:
$NAME$CRT"
Exit
fi
echo "Client’s cert found: $NAME$CR"
#Then, verify that there is a private key for #that client
if [ ! -f $NAME$KEY ]; then
echo "[ERROR]: Client 3des Private Key not found:
$NAME$KEY"
Exit
fi
echo "Client’s Private Key found: $NAME$KEY"
#Confirm the CA public key exists if [ ! -f $CA ]; then
echo "[ERROR]: CA Public Key not found: $CA"
exit
fi
echo "CA public Key found: $CA"
#Confirm the tls-auth ta key file exists if [ ! -f $TA ];
then
echo "[ERROR]: tls-auth Key not found: $TA"
exit
fi
echo "tls-auth Private Key found: $TA"
#Ready to make a new.opvn file - Start by #populating with
the default file
cat $DEFAULT > $NAME$FILEEXT
#Now, append the CA Public Cert echo "<ca>" >> $NAME$FILEEXT
cat $CA >> $NAME$FILEEXT
_16
echo "</ca>" >> $NAME$FILEEXT
#Next append the client Public Cert
echo "<cert>" >> $NAME$FILEEXT
cat $NAME$CRT | sed -ne '/-BEGIN CERTIFICATE-/,/-END
CERTIFICATE-/
p' >> $NAME$FILEEXT
echo "</cert>" >> $NAME$FILEEXT
#Then, append the client Private Key
echo "<key>" >> $NAME$FILEEXT
cat $NAME$KEY >> $NAME$FILEEXT
```

```
echo "</key>" >> $NAME$FILEEXT
#Finally, append the TA Private Key
echo "<tls-auth>" >> $NAME$FILEEXT
cat $TA >> $NAME$FILEEXT
echo "</tls-auth>" >> $NAME$FILEEXT
echo "Done! $NAME$FILEEXT Successfully Created."
#Script written by Eric Jodoin
\ No newline at end of file
```

Como se ha realizado antes, hay que darle permisos al propietario del fichero, para ello ejecutar el siguiente comando en el directorio keys:

```
cd /etc/openvpn/easy-rsa/keys/
chmod 700 MakeOVPN.sh
```

Finalmente se ejecuta el script con el siguiente comando:

```
./MakeOVPN.sh
```

Cuando se ejecuta el script se piden los nombres de los certificados que se crearon anteriormente. En este caso, client1. Si todo va bien se muestra el siguiente mensaje:

```
Done! client1.ovpn Successfully Created.
```

Se debe repetir este paso para todos los clientes creados.

Finalmente, se deben descargar con el protocolo SCP (Security Copy Protocol) de la Raspberry Pi los archivos con extensión .ovpn que han sido creados por cada cliente para utilizarlos en el programa cliente y poder conectarse así al servidor VPN. Para ello se puede utilizar aplicaciones como WinSCP en Windows que es de las más utilizadas o Filezilla [48] que vale para todas las plataformas.

Puede ocurrir que no se tenga permiso para acceder a la carpeta openvpn que es el directorio donde se encuentran los archivos con extensión .ovpn. Para ello se debe dar permisos de lectura y escritura a la carpeta y una vez copiados, se vuelven a eliminar los permisos para que solo tenga acceso el propietario:

```
chmod 777 -R /etc/openvpn
chmod 600 –R /etc/openvpn
```

Una vez realizados todos estos pasos, ya se tiene funcionando el servidor VPN y creados los clientes que se van a conectar. Para conectarse al servidor VPN con el servicio openvpn se puede utilizar la interfaz gráfica de usuario OpenVPN GUI [33] para Windows como se explica en el anexo de este documento.

Capítulo 5: Snort y su implementación en una plataforma Raspberry Pi

En este capítulo se va a desarrollar de una manera didáctica la solución elegida para analizar el tráfico de la red que pasa por la VPN implementada.

En primer lugar, se va a explicar que es la herramienta Snort y cuál es su finalidad. A continuación, se va a desarrollar su implementación y las configuraciones que se han de tener en cuenta.

5.1. Definición de Snort

Snort [34] está disponible bajo licencia GPL, gratuito y funciona bajo plataformas Windows y UNIX/Linux, es uno de los productos de IDS/IPS más populares.

Lo característico de Snort es que permite controlar todos los paquetes que atraviesan la red en la cual se ha instalado, es altamente configurable debido a que dispone de una gran cantidad de filtros o patrones ya predefinidos, así como actualizaciones constantes ante casos de ataques, barridos o vulnerabilidades y usa reglas programadas por el usuario para detectar y/o filtrar paquetes. Una regla Snort es un parámetro personalizado que le dice a Snort que busque tráfico específico y realice una acción cuando se detecte.

Funciona como sniffer (se puede comprobar en tiempo real que ocurre en la red mediante la consola de comandos), registro de paquetes (permite guardar en un archivo los logs para su posterior análisis, un análisis offline) o como un IDS normal (en este caso NIDS).

El objetivo por tanto es configurar un laboratorio Snort en el que se configuren reglas para detectar posibles ataques o tráfico malicioso que pase por la VPN.

5.2. Implementación de Snort

Antes de comenzar con la implementación de Snort se deben de haber seguido las configuraciones iniciales de la Raspberry Pi explicada en los Anexos, destacar sobre todo expandir el espacio de memoria de la tarjeta SD una vez que Rapsbian esté instalado, comprobar que la Rapsberry Pi esta actualizada con el comando sudo apt-get update && sudo apt-get dist-upgrade -y, asignar una IP estática para la Raspberry Pi, en este caso, 192.168.1.30 y es recomendable realizar un backup de la imagen de la Raspberry Pi por motivos de seguridad.

Una vez se tengan configurados los aspectos más importantes se puede comenzar con la implementación de Snort en la plataforma.

Para realizar la instalación se puede realizar desde la propia terminal de la Raspberry Pi o de manera más cómoda mediante SSH con el comando ssh pi@IPEstaticaRaspberryPi.

Se debe descargar el código fuente de Snort de la página principal snort.org en la sección "Get Snort" (*Figura 5.1*) donde se debe verificar la versión del código antes de descargarlo. Para facilitar la administración y localización del mismo se ha creado un directorio y se ha convertido en el directorio de trabajo actual.

```
mkdir -p ~ /sourcecode/snort_src/
cd ~ /sourcecode/snort_src /
```

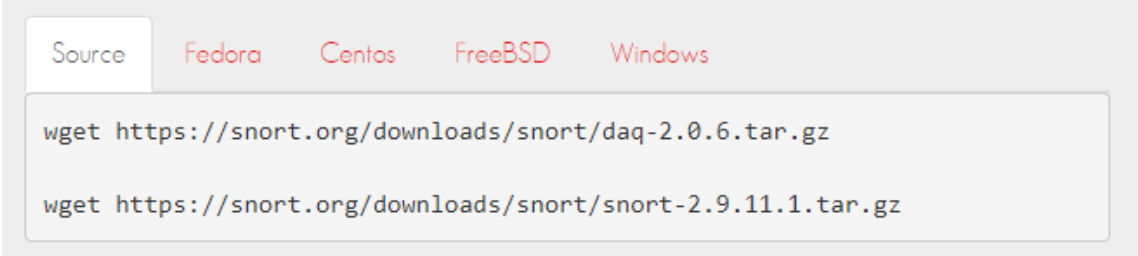


Figura 5.1. Página principal de Snort.

El primer enlace del código fuente a descargar es la Biblioteca de Adquisición de Datos (DAQ), que es una pieza de código de la que depende Snort para ejecutarse. El segundo enlace es el código fuente de Snort en sí.

5.2.1. Instalación de la dependencia DAQ

En primer lugar, se debe descargar el archivo en el directorio de trabajo actual teniendo en cuanta la última versión, se debe comprobar en la página principal de Snort.

```
wget https://snort.org/downloads/snort/daq-2.0.6.tar.gz
```

Se debe extraer el archivo ya que se ha descargado un archivo tar comprimido:

```
tar xvfz daq-2.0.6.tar.gz
```

Para comenzar a ejecutar los scripts que permiten el instalar el software hay que desplazarse al directorio donde se encuentran descargados:

```
cd daq-2.0.6 /
```

Es necesario descargar e instalar previamente unas dependencias para compilar el código y tres paquetes de software con las que cuenta el administrador de paquetes de Raspbian, se deben lanzar los siguientes comandos:

```
sudo apt-get install bison -y
sudo apt-get install flex -y
```

```
sudo apt-get install libpcap-dev -y
sudo apt-get install libpcre3-dev -y
sudo apt-get install libdumbnet-dev -y
```

Una vez se hayan instalado todas las dependencias necesarias se puede compilar el DAQ. Con el paquete checkinstall será más fácil, para ello:

```
sudo apt-get install checkinstall
```

Ahora se puede comenzar a compilar e instalar el código. Ejecute los siguientes comandos:

```
./configure
make
sudo checkinstall -D --install=no --fstrans=no
```

Con el último comando se crea un paquete con la extensión .deb, se debe dar una descripción, por ejemplo "snort-daq", y dejar el resto de los valores predeterminados. Por último, se instala el DAQ con el comando dpkg que permite analizar o instalar paquetes .deb:

```
sudo dpkg -i daq_2.0.6-1_armhf.deb
```

5.2.2. Instalación de Snort

Una vez instalada la dependencia DAQ, se procede a instalar Snort, para ello se regresa al directorio creado y se ejecuta el comando para descargar el código fuente que se especifica en la página oficial de Snort (*Figura 5.1*). De nuevo comprobar la última versión oficial.

```
wget https://snort.org/downloads/snort/snort-2.9.11.1.tar.gz
tar xvfz snort-2.9.11.1.tar.gz
cd ~ /sourcecode/snort_src/snort-2.9.11.1
```

Una vez dentro del directorio de origen, comienza el proceso de compilación e instalación como se realizó anteriormente.

```
./configure --enable-sourcefire
make
sudo checkinstall -D --install=no --fstrans=no
```

Con el último comando se crea un paquete con la extensión .deb, se debe dar una descripción, por ejemplo "snort", y dejar el resto de los valores predeterminados. Por último, se instala Snort con el comando dpkg que permite analizar o instalar paquetes .deb:

```
sudo dpkg -i snort_2.9.11.1-1_armhf.deb
```

Es necesario actualizar el caché de la biblioteca compartida, para ello ejecutar el siguiente comando:

```
sudo ldconfig
```

El siguiente comando permite verificar si Snort se ha instalado en la ubicación usr /local/bin /location.

```
which snort
```

Snort se ejecutará desde la ubicación usr/local/bin. Se va a agregar un enlace simbólico en /usr/sbin que apunte a /usr/local/bin/snort para que el sistemaprueda ubicar el binario de Snort a la hora de crear el usuario y grupo de snort.

```
sudo ln -s /usr/local/bin/snort /usr/sbin/snort
```

Finalmente verificar si Snort funciona en la versión correcta (*Figura 5.2*):

```
snort –version
```

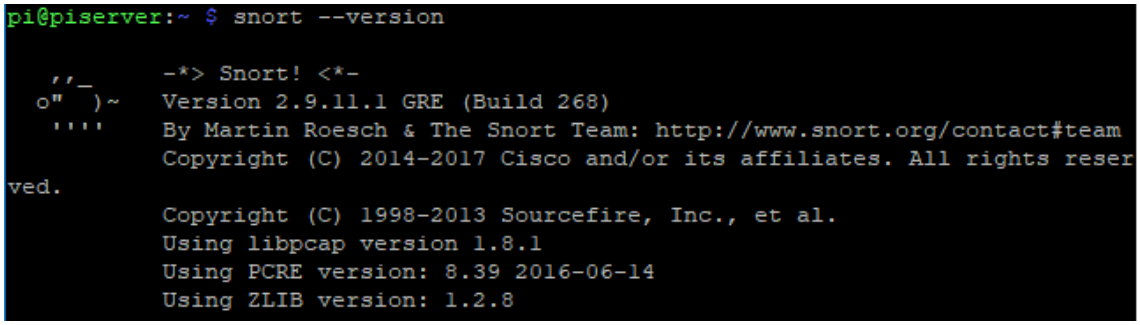


Figura 5.2. Versión de Snort instalada.

A continuación, se va a crear un usuario y un grupo de Snort para que que Snort no se ejecute como root. Este usuario ejecutará el proceso.

Una vez instalado Snort se crea un usuario para que Snort no se ejecute como usuario root.

```
sudo groupadd snort
sudo useradd snort -r -s /sbin/nologin -c SNORT_IDS -g snort
```

Ahora se van a crear los directorios que contendrán los archivos de configuración:

```
sudo mkdir -p /etc/snort/rules/iplists
sudo mkdir /etc/snort/preproc_rules
sudo mkdir /usr/local/lib/snort_dynamicrules
sudo mkdir /etc/snort/so_rules
```

Con estos directorios creados, se van a añadir archivos de reglas vacías para Snort que sirven para definir las reglas de Snort.

```
sudo touch /etc/snort/rules/local.rules
sudo touch /etc/snort/rules/iplists/white_list.rules
sudo touch /etc/snort/rules/iplists/black_list.rules
```

Snort también necesitará una carpeta para generar archivos de registro

```
sudo mkdir /var/log/snort
```

El usuario y el grupo de Snort no podrán escribir archivos de registro o leer la configuración sin modificar los privilegios de los directorios que se han creado.

```
sudo chmod -R 5775 /etc/snort
sudo chmod -R 5775 /var/log/snort
sudo chmod -R 5775 /usr/local/lib/snort_dynamicrules
sudo chown -R snort:snort /etc/snort
sudo chown -R snort:snort /var/log/snort
```

```
sudo chown -R snort:snort /usr/local/lib/snort_dynamicrules
```

Solo quedan unas últimas configuraciones, se debe volver al directorio /etc donde se extrajeron los archivos de configuración de Snort y copiar la configuración relacionada con Snort a la carpeta de configuración de Snort creada automáticamente al instalar el IDS.

```
cd ~/sourcecode/snort_src/snort-2.9.11.1/etc
sudo cp *.conf /etc/snort
sudo cp *.config /etc/snort
sudo cp *.map /etc/snort
```

5.2.3. Configuración y ejecución de Snort

El comportamiento de Snort se establece a partir de un archivo de configuración que responde al nombre de snort.conf. En este archivo se especifican las distintas opciones que delimitan como se comportará Snort y de qué forma trabajará. Este archivo se encuentra en el directorio /etc/snort.

Se ha accedido al fichero y se han comentado todas las líneas que contienen "include $RULE_PATH" ya que no se van a utilizar. Tras esto, se han realizado otras configuraciones en el mismo fichero.

Modificar la línea ipvar HOME_NET con su propia subred IP seguida de un sufijo/CIDR que permite representar una subred (255.255.255.0).

```
sudo nano /etc/snort/snort.conf
ipvar HOME_NET 192.168.0.0/24
```

La siguiente configuración es modificar la ruta de los ficheros donde se van a almacenar las reglas que corresponden con los ficheros creados anteriormente (*Figura 5.3*).

```
var RULE_PATH /etc/snort/rules
var SO_RULE_PATH /etc/snort/so_rules
var PREPROC_RULE_PATH /etc/snort/preproc_rules
var WHITE_LIST_PATH /etc/snort/rules/iplists
var BLACK_LIST_PATH /etc/snort/rules/iplists
```

```
# such as:  c:\snort\rules
var RULE_PATH /etc/snort/rules
var SO_RULE_PATH /etc/snort/so_rules
var PREPROC_RULE_PATH /etc/snort/preproc_rules

# If you are using reputation preprocessor set these
# Currently there is a bug with relative paths, they are relative to where snort is
# not relative to snort.conf like the above variables
# This is completely inconsistent with how other vars work, BUG 89986
# Set the absolute path appropriately
var WHITE_LIST_PATH /etc/snort/rules/iplists
var BLACK_LIST_PATH /etc/snort/rules/iplists
```

Figura 5.3. Modificación del fichero snort.conf.

A continuación, en la sección del fichero de reglas específicas se debe eliminar el comentario de la primera línea para que se lea “include $RULE_PATH/local.rules”, guardar el fichero de configuración y ejecutar el siguiente comando para comprobar si se ha realizado una correcta configuración y no ha habido ningún error:

```
sudo snort -T -c /etc/snort/snort.conf -i eth0
```

Para controlar el tráfico que pasa por la red, se pueden crear reglas específicas en el archivo local.rules ejecutando el comando sudo nano /etc/snort/rules/local.rules.

Se debe volver a ejecutar el comando sudo snort -T -c /etc/snort/snort.conf -i eth0 para comprobar que todo ha ido bien y finalmente ejecutar Snort con el siguiente comando:

```
sudo snort -A console -u snort -g snort -c /etc/snort/snort.conf -l
/var/log/snort -i eth0
```

El parámetro -A indica que la salida de la detección de Snort sea por consola, -u y -g permiten seleccionar el usuario y grupo a ejecutar, -c permite ejecutar un archivo de configuración en concreto, -l permite registrar los logs en un archivo específico y -i permite detectar el tráfico en la interfaz de ethernet indicada (*Figura 5.4*).

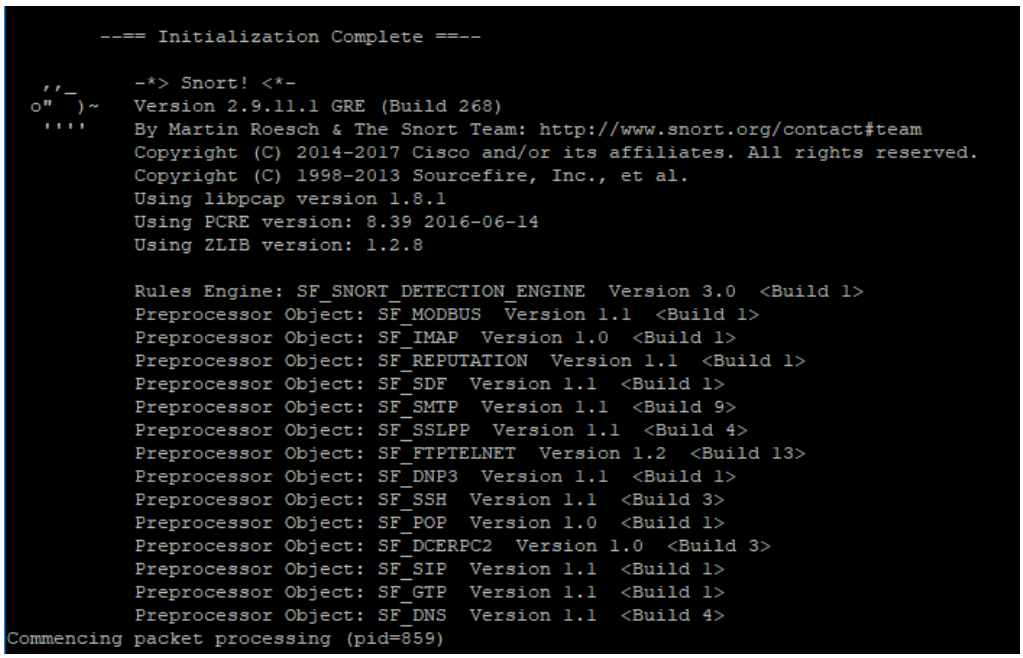


Figura 5.4. Ejecución de la herramienta Snort.

Capítulo 6: Validación del proyecto de seguridad

En este capítulo se va a validar el funcionamiento del servidor VPN en redes públicas utilizando el sniffer Wireshark. La validación consiste en capturar paquetes y comprobar si se encripta el tráfico con el protocolo openVPN.

Una vez se demuestre que es factible utilizar una Raspberry Pi como una VPN en una red corporativa de carácter medio, se explicarán las reglas implementadas en la herramienta Snort para analizar el tráfico y qué tipo de ataques es capaz de detectar.

6.1. Utilización de Wireshark

Wireshark es un analizador de protocolos que permite ver el tráfico que pasa a través de una red. Se ha utilizado este sniffer ya que se puede utilizar como una herramienta didáctica y es de los más conocidos. Permite analizar la información capturada y filtrar los paquetes mediante diferentes categorías, como por ejemplo por una IP específica, tipo de protocolo, etc.

En este caso, para la validación de la solución se ha realizado la conexión a una red Wi-Fi abierta, concretamente un restaurante. La forma más rápida de comprobar si la solución funciona es validarse con usuario y contraseña en una página que no utilice el protocolo https, ya que va en texto plano y es totalmente visible y accesible a cualquiera. Se ha utilizado una página cualquiera que no utiliza el protocolo https, en concreto http://laspaginasverdes.com/, se ha procedido el registro en la misma y se ha realizado el login y navegación por ella. En las siguientes dos secciones se explica más en detalle las conexiones con y sin VPN a la página web comentada.

6.1.1. Conexión sin VPN

En esta prueba, no se ha habilitado la conexión VPN con OpenVPN GUI, directamente se ha realizado la conexión con la red Wi-Fi del restaurante. Tras esto, se ha procedido a capturar paquetes con Wireshark mientras se interactuaba con la página.

Como se muestra en la *Figura 6.1*, tras capturar los paquetes cuando se interactuaba con la página, se ha filtrado peticiones HTTP con el método POST mediante la siguiente sentencia http.request.method == “POST”. Se encuentra un paquete que coincide con el inicio de sesión en la página web, por lo que se ha analizado y comprobado su contenido pulsando botón derecho sobre el mismo paquete y seleccionando la opción “Follow TCP Stream”.

Una vez seleccionada la opción “Follow TCP Stream” se muestra lo que se ve en la *Figura 6.2*. De este cuadro de diálogo se puede comprobar que tanto el nombre de

usuario “nazaret_pruebas” como la contraseña “nazaretpruebas”, que se encuentra resaltado en la imagen, está en texto plano y cualquier usuario con un mínimo de conocimientos de esta herramienta podría obtener esta información.

6.1.2. Conexión con VPN

En esta prueba se ha habilitado la conexión VPN con OpenVPN GUI, y tras esto se ha accedido a la página web del restaurante y se han realizado las mismas pruebas que en la sección *6.1.1. Conexión sin VPN*.

El resultado de la captura de los paquetes ha resultado todo lo contrario que la prueba anterior, estos se encuentran encriptados y no se puede ver el contenido de los mismos ni el nombre de usuario y contraseña ya que utiliza el protocolo OpenVPN como se muestra en la *Figura 6.3*. Esto se debe a que, al conectarse a una VPN, el software crea una interfaz de red virtual y le asigna una dirección IP. Los paquetes enrutados sobre esa interfaz se envían al cliente VPN, que los encripta y los envía dentro de los nuevos paquetes al servidor VPN, que luego se enrutan al dispositivo Wi-Fi real. Funciona igual que el protocolo HTTPS, en el sentido de que utiliza un programa cliente que autentica la clave del servidor con un certificado de confianza para iniciar una sesión cifrada, sobre la que se envía todo el tráfico. El único agujero conocido en el proceso radica en desconfiar de un certificado y en este caso ha sido creado en la propia implementación del servidor, por lo que es totalmente de confianza.

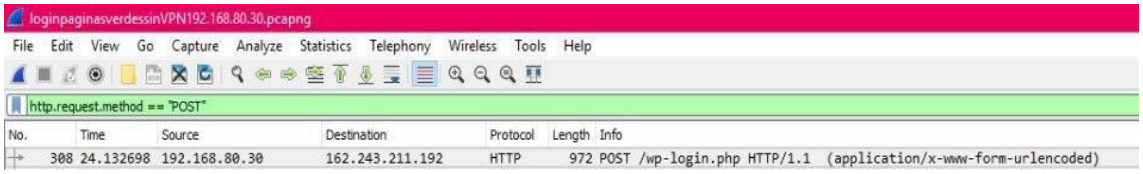


Figura 6.1. Filtrado de paquetes en el sniffer Wireshark.

Figura 6.2. Detalles de un paquete en el sniffer Wireshark.

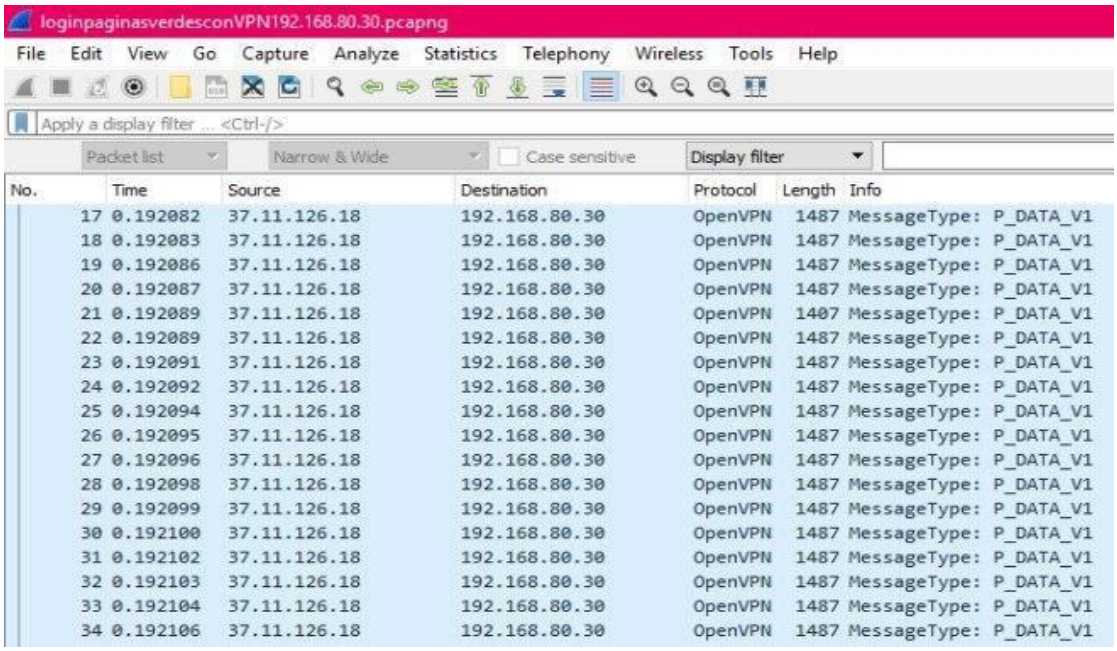


Figura 6.3. Paquetes cifrados en el sniffer Wireshark.

6.2. Reglas de Snort

En este apartado se van a desarrollar las reglas implementadas en Snort para poder detectar anomalías en el tráfico de la red además de dar una descripción sobre los diferentes tipos de ataques. Todas las reglas comentadas a continuación han sido añadidas al fichero /etc/snort/rules/local.rules.

6.2.1. NMAP

La herramienta nmap es capaz de utilizar diferentes técnicas de evasión para determinar los hosts que se están ejecutando, los servicios que utilizan, las versiones de estos... Soporta escaneos sobre ciertos puertos específicos, entre rangos IP específicos, uso se paquetes NULL, FIN, XMAS y ACK, además SYN que es el paquete por defecto.

- Escaneo NMAP Ping

Lo normal es que un atacante comience un ataque identificando el estado del host enviando un paquete ICMP para realizar un escaneo mediante ping. Por lo tanto, la primera regla de Snort a comentar es una regla que analice el análisis NMAP ping cuando alguien intente escanear la red para identificar el estado del host.

```
alert icmp any any -> 192.168.1.30 any (msg: "Escaneo NMAP Ping";
dsize: 0; sid: 10000004; rev: 1;)
```

Cuando el atacante ejecute el comando `nmap -sP 192.168.1.30 -disable-arp-ping`, saltará el mensaje especificado en la regla "Escaneo NMAP ping ".

El objetivo, por tanto, es detectar la IP del atacante que está realizando el análisis y como contramedida bloquear esta IP en la red corporativa y proteger la red de futuros ataques.

- Escaneo NMAP TCP y UDP

Para conectarse con la red objetivo, el atacante puede ir con la enumeración de redes ya sea usando el protocolo TCP o el protocolo UDP, para detectar este ataque se han creado las siguientes reglas:

```
alert tcp any any -> 192.168.1.30 2222 (msg: "Escaneo NMAP TCP "; sid:10000005; rev:2;)

alert udp any any -> 192.168.1.30 any (msg: "Escaneo NMAP UDP"; sid:10000005; rev:2;)
```

Se puede especificar que la regla haga referencia a un puerto en concreto, como en la regla TCP que es específica para el puerto 2222, o para todos los puertos como en le regla UDP.

Cuando el atacante ejecute el comando nmap `-sT -p2222 192.168.1.30`, saltará el mensaje especificado en la regla “Escaneo NMAP TCP” y cuando ejecute el comando `nmap -sU -p68 192.168.1.105`, saltará el mensaje “Escaneo NMAP UDP”

Existen diferentes variantes de reglas por ejemplo mediante TCP en función de la conexión o flag utilizado por el atacante:

- Escaneo NMAP XMAS

En lugar de utilizar SYN, SYN / ACK, el atacante utiliza un escaneo XMAS para conectarse con el objetivo enviando paquetes de datos a través de indicadores FIN, PSH y URG activados. Si el puerto está abierto no hay respuesta, pero si el puerto está cerrado el destinatario responde con un paquete RST/ACK.

```
alert tcp any any -> 192.168.1.30 2222 (msg: "Exploracion del arbol NMAP XMAS"; flags:FPU; sid:1000006; rev:1;)
```

El código ejecutado por el atacante sería `nmap -sX -p2222 192.168.1.30`, y se mostraría el mensaje “Exploracion del arbol NMAP XMAS”

- Escaneo NMAP FIN

Un escaneo FIN es similar a un escaneo XMAS pero envía un paquete con el flag FIN activo, recibe las mismas respuestas que XMAS.

```
alert tcp any any -> 192.168.1.30 2222 (msg: "Escaneo NMAP FIN"; flags:F; sid:1000008; rev:1;)
```

El código ejecuta por el atacante sería nmap -sF -p2222 192.168.1.30, y se mostraría el mensaje “Escaneo NMAP FIN”.

- Escaneo NMAP NULL

El escaneo NULL es similar a los escaneos XMAS y FIN, pero envía un paquete con ningún *flag* activo.

```
alert tcp any any -> 192.168.1.30 2222 (msg: "Escaneo NMAP NULL"; flags:0; sid:1000009; rev:1;)
```

El código ejecuta por el atacante sería `nmap -sN -p2222 192.168.1.30`, y se mostraría el mensaje “Escaneo NMAP NULL”.

6.2.2. Inyección SQL

Aunque en principio este tipo de ataques esta orientado a obtener información de bases de datos y páginas web aprovechando vulnerabilidades o fallos en la implementación del código se han añadido las siguientes reglas al fichero.

```
alert tcp any any -> 192.168.1.30 any (msg:"Posible ataque SQL injection – Comentarios y delimitador de texto"; flow:to_server; pcre:"/(\%27)|(\')|(\-\-)|(%23)|(#)/i"; sid:1000001; rev:1;)
```

```
alert tcp any any -> 192.168.1.30 any (msg:"SQL Injection - SQL key word"; flow:to_server;
pcre:"/((\%3D)|(=)).*((\%6F)|o|(\%4F))((\%72)|r|(\%52))/i";
sid:1000002;
rev:1;)
```

```
alert tcp any any -> 192.168.1.30 any (msg:"SQL Injection - SQL key word";flow:to_server;pcre:"/((\%3D)|(=)).*((\%61)|a|(\%41))((\%6E)|n|(\%4E))((\%64)|d|(\%44))/i"; sid:1000005; rev:1;)
```

donde flow: to_server son peticiones de servidores A hacia B y pcre permite la utilización de expresiones regulares como es el caso de detección de comentarios y delimitador utilizados para realizar ataques de inyección SQL.

6.2.3. DoS

Un ataque Dos (Denial of Service) es un ataque de denegación de servicio que causa que un servicio o recurso sea inaccesible a los usuarios tras realizar un gran número de peticiones por parte del atacante o botnets. Un ataque DoS puede ser perpetrado de varias formas:

- Consumo de recursos computacionales.
- Modificación de información de configuración.
- Modificación de información de estado (como sesiones TCP).
- Interrupción de componentes físicos de red.
- Impedimente de comunicación entre usuarios de un servicio y la víctima.

```
alert tcp any any -> 192.168.1.30 any (msg:"Posible ataque TCP DoS"; flow:
to_server; flags: S; detection_filter:track by_src, count 50, seconds 1;
sid:1000003;rev:1;)
```

```
alert ICMP any any -> 192.168.1.30 any (msg:"Posible ataque ICMP DoS";
detection_filter: track by_src, count 50, seconds 1; sid:1000004; rev:1; )
```

El parámetro detection_filter define una tasa que debe ser excedida por un host de origen o de destino antes de que una regla pueda generar un evento. El formato de detection_filter es el siguiente [39]:

- track by_src: es la tasa se rastrea por dirección IP de origen. Esto significa que el conteo se mantiene para cada dirección IP de origen única
- count 50, seconds 1: Si el número de peticiones en un segundo llega a las 50 saltará una alerta como posible ataque DoS.

6.2.4. Ataque de fuerza Bruta

Un ataque de fuerza bruta consiste en intentar averiguar una contraseña probando todas las combinaciones posibles hasta obtener la correcta. Lo más importante ante este tipo de ataques es generar contraseñas robustas que no sean predictibles y seguras. Aun así, es recomendable crear una regla que compruebe si se está realizando este tipo de ataque, para ello se ha implementado la siguiente regla:

```
alert tcp any any -> 192.168.1.30 22 ( \ msg:”Posible ataque de fuerza bruta mediante SSH”; flow:established,to_server; \ content:”SSH”; nocase; offset:0; depth:4; \ detection_filter: track by_src, count 30, seconds 60; \ sid:1000001; rev:1;)
```

Esta regla saltará en cada intento de inicio de sesión fallido desde cualquier IP durante un período de muestreo de 60 segundos, después de los primeros 30 intentos fallidos de inicio de sesión. Puede saltar como alerta alert o directamente se puede bloquear con el parámetro drop.

6.2.5 Otras reglas

Otra regla interesante que implementar es si se tiene alguna carpeta o contenido que se quiera comprobar si ha sido vulnerado para ello se ha creado la regla siguiente donde en el parámetro content se puede añadir cualquier archivo:

```
alert tcp 192.168.1.130 any -> any any (msg:”Command Shell Access”; content: "C:\\Users\\Administrator\\Desktop\\prueba.txt"; sid:1000004; rev:1;)
```

Una regla que permite comprobar si un usuario Administrador del sistema se ha conectado, para ello se utilizan expresiones regulares:

```
alert tcp any any > any 21 (msg: “Usuario Administrador conectado”; content: “USER”; pcre: “/administrator/i“;)
```

Detectar el cambio entre directorios de ciertos ejecutables:

```
alert tcp any any > any any (msg: “Alerta movimiento de ejecutables “; pcre: “/(\.exe|\.bat)/i“;)
```

Se pueden crear diferentes reglas en función de la especificación del cliente, existen paquetes de reglas por defecto, pero suelen ser más generales, por ello se han creado las reglas específicas anteriores. Destacar que el mejor enfoque para cualquier organización es realizar pruebas de penetración para la aplicación, escribir reglas SNORT para proteger contra los ataques y modificar el código para la implementación adecuada de la seguridad.

6.3. Validación del IDS Snort

Una vez añadidas las reglas en el fichero correspondiente se han realizado peticiones al host para comprobar que el sistema de detección de intrusos funciona correctamente (*Figura 6.4).*

Se han realizado escaneos desde diferentes equipos y han saltado las reglas como se muestra en la *Figura 6.5* y en la *Figura 6.6*. En este caso se ha hecho que la información de las peticiones se muestre por consola, pero es configurable para almacenarlo en logs, enviar las peticiones a un correo específico, etc.

Con esta información se podría bloquear las peticiones del equipo 37.11.124.137 y 192.168.1.135. para evitar así posibles ataques de IPs sospechosas.

```
#NMAP

#NMAP PING
alert icmp any any -> 192.168.1.30 any (msg: "Escaneo NMAP Ping"; dsize: 0; sid: 10000004; rev: 1;)
#NMAP TCP/UDP
alert tcp any any -> 192.168.1.30 2222 (msg: "Escaneo NMAP TCP "; sid:10000005; rev:2;)
alert udp any any -> 192.168.1.30 any (msg: "Escaneo NMAP UDP"; sid:10000005; rev:2;)
#NMAP XMAS
alert tcp any any -> 192.168.1.30 2222 (msg: "Exploracion del arbol NMAP XMAS"; flags:FPU; sid:1000006; rev:1;
#NMAP FIN
alert tcp any any -> 192.168.1.30 2222 (msg: "Escaneo NMAP FIN"; flags:F; sid:1000008; rev:1;)
#NMAP NULL
alert tcp any any -> 192.168.1.30 2222 (msg: "Escaneo NMAP NULL"; flags:0; sid:1000009; rev:1;)

#SQL INJECTION
alert tcp any any -> 192.168.1.30 any (msg:"Posible ataque SQL injection - Comentarios y delimitador de texto (Comme$
alert tcp any any -> 192.168.1.30 any (msg:"SQL Injection - SQL key word"; flow:to_server; pcre:"/((\%3D)|(=)).*((\%$
rev:1;)
alert tcp any any -> 192.168.1.30 any (msg:"SQL Injection - SQL key word"; flow:to_server;pcre:"/((\%3D)|(=)).*((\%6$

#DoS
alert tcp any any -> 192.168.1.30 any (msg:"Posible ataque TCP DoS"; flow: to_server; flags: S; detection_filter:tra$
alert ICMP any any -> 192.168.1.30 any (msg:"Posible ataque ICMP DoS"; detection_filter: track by_src, count 50, sec$

#BRUTE FORCE ATTACK
alert tcp anyany > 192.168.1.30 22 ( \ msg:"Posible ataque de fuerza bruta mediante SSH"; flow:established,to_server$

#CONTENT
alert tcp 192.168.1.130 any -> any any (msg:"Command Shell Access"; content: "C:\\Users\\Administrator\\Desktop\\pru$

#ADMIN
alert tcp any any > any 21 (msg: "Usuario Administrador conectado"; content: "USER"; pcre: "/administrator/i";)

#EXECUTABLES
alert tcp any any > any any (msg: "Alerta movimiento de ejecutables "; pcre: "/(\.exe|\.bat)/i";)
```

Figura 6.4. Reglas de Snort.

```
Commencing packet processing (pid=1030)
06/01-11:53:25.104630  [**] [1:10000005:2] Escaneo NMAP UDP [**] [Priority: 0] {UDP} 37.11.124.137:54589 -> 192.168.1
.30:137
```

Figura 6.5. Regla "Escaneo NMAP UPD".

```
06/01-12:19:46.541663  [**] [1:10000004:1] Escaneo NMAP Ping [**] [Priority: 0] {ICMP} 192.168.1.135 -> 192.168.1.30
```

Figura 6.6. Regla "Escaneo NMAP Ping".

Capítulo 7: Conclusiones.

En este capítulo, se van a recoger las conclusiones que se han obtenido tras terminar el proyecto, así como el trabajo futuro que se deberá completar, para ir elevando el nivel de seguridad en la plataforma y añadir nuevas funcionalidades.

5.1. Conclusiones generales

Tras realizar el proyecto se ha obtenido un sistema de seguridad que permite a una empresa de carácter medio securizar las conexiones de los usuarios mediante el cifrado de estas además de poder detectar cualquier anomalía en el tráfico de la red.

Se demuestra, por tanto, que con poco dinero y una base de programación/desarrollo se es capaz de montar un sistema totalmente confiable y útil.

5.2. Conclusiones específicas

En este trabajo se ha analizado la plataforma Raspberry Pi, entre otras, como posible herramienta orientada a proyectos relacionados con la seguridad y se han obtenido resultados muy positivos. La plataforma debido a sus características técnicas en hardware y software como en la potencia de la misma y la gran comunidad de apoyo con la que cuenta es una solución que permite plantear proyectos prácticos en cualquier área de la ciberseguridad con diferentes complejidades.

Uno de los problemas de avanzar en un proyecto o aprendizaje es a la hora de intentar utilizar o adquirir nuevas infraestructuras o herramientas además del coste económico que suponen. Por ello, se han analizado varias plataformas de muy bajo coste para que no sea un problema a la hora de llevar este proyecto a cabo ya que el precio es bastante asequible y con la Raspberry Pi se obtiene incluso un ordenador que puede realizar tareas simples. Al fin y al cabo, las empresas buscan una solución eficaz y con el menor coste posible.

Lo único a destacar como requisito para poder realizar este proyecto es tener unas mínimas nociones de informática y del sistema operativo Linux. Es por ello por lo que a lo largo del trabajo y en los apéndices se han explicado ciertos conceptos de una manera más específica y didáctica.

5.3. Líneas de trabajo futuro

Este trabajo ha cumplido los objetivos planteados en el Capítulo 1, pero tras la realización de los mismos, se abren interesantes líneas de trabajo futuro, como aumentar capas de seguridad en la plataforma o realizar nuevas funcionalidades.

5.3.1. Capas de seguridad

Se pueden añadir mejoras a la plataforma para aumentar la seguridad en la misma, una posible solución para añadir una capa más es la combinación de TOR (The Onion Router) con la red VPN. La utilización de TOR permite un encubrimiento de la IP original, cifrado de tráfico, etc., ya que la información del usuario rebota entre diferentes nodos antes de llegar al destino final. Aunque las ventajas que ofrece son similares a las que realiza la VPN, se aumenta la seguridad del anonimato a la hora de navegar por Internet y más si se utilizan conexiones seguras (HTTPS). Es recomendable combinar ambos si la VPN es fiable, y en este caso lo es, porque se ha implementado desde cero.

Por otra parte, hay que destacar que el uso de ambos disminuye la velocidad de navegación, ya que TOR es mucho más lento que VPN y debido a que el nodo final desencripta los datos para acceder al destino indicado por el usuario, abre la posibilidad a vulnerabilidades.

5.3.2. Nuevas funcionalidades

La implementación de la red VPN en la Raspberry Pi se puede utilizar en diferentes contextos, tanto en un usuario doméstico preocupado por la seguridad de sus redes, en grandes empresas para que se realicen conexiones de una manera segura, como en coches conectados a Internet.

Una posible línea de trabajo para el futuro sería orientar este proyecto en contextos de aplicación para fábricas, ciudades inteligentes o coches conectados utilizando una red VPN desde el coche hasta un servidor proxy aislando así las conexiones y poder securizar un único punto de entrada.

Otra posible idea de línea de trabajo sería poder establecer una conexión segura al conectarse al Wi-Fi de un avión a miles de metros de altura con la reconfiguración del servidor VPN con OpenVPN instalado como se ha hecho en este proyecto.

Otra sencilla línea de trabajo posible es la configuración de Pi-Hole en la Raspberry Pi permitiendo así el bloqueo de anuncios y trackers, es posible implementarse a gran escala como en oficinas o empresas.

Por ello, se pueden realizar diferentes proyectos con un servidor VPN implementado en una Raspberry Pi y seguir interesantes líneas de trabajo ya que las posibilidades de crear y mejorar diferentes aspectos de la misma son infinitas.

Bibliografía

[1] INCIBE, «Instituto Nacional de Tecnologías de la Comunicación,» [En línea]. Available: https://www.incibe.es/extfrontinteco/img/File/demostrador/monografico_catalogo_seguridad_perimetral.pdf.

[2] P. Turmero, «Monografias.com,» [En línea]. Available: http://www.monografias.com/trabajos106/elementos-basicos-seguridad-perimetral/elementos-basicos-seguridad-perimetral.shtml.

[3] Avaya, «Avaya,» 27 Septiembre 2016. [En línea]. Available: http://www.avaya.com/es/about-avaya/newsroom/news- releases/2016/esp_np_mx_20160927/.

[4] INCIBE, «Instituto Nacional de Ciberseguridad de España,» 20 Octubre 2015. [En línea]. Available: https://www.certsi.es/blog/divide-venceras-segmentacion-rescate.

[5] URJCx, «Ciberseguridad: Ataques y Contramedidas,» 17 Octubre 2016. [En línea]. Available: https://urjcx.urjc.es/courses/course-v1:URJCx+URJCx104+IAx002/about.

[6] A. Networking, «Avaya,» 27 Septiembre 2016. [En línea]. Available: https://www.avaya.com/es/about-avaya/newsroom/news-releases/2016/esp_np_mx_20160927/.

[7] M. Beltrán, «Unidad 7: Contramedidas de red/protocolo,» [En línea].

[8] C. Borghello, «Segu-info,» [En línea]. Available: http://www.segu-info.com.ar/firewall/firewall.htm.

[9] «Ymant,» 14 Diciembre 2016. [En línea]. Available: http://www.ymant.com/dmz-zona-desmilitarizada/.

[10] J. García, «Josedomingo.org,» 2016. [En línea]. Available: http://plataforma.josedomingo.org/pledin/pluginfile.php/696/mod_resource/content/0/UOC_cortafuegos.pdf.

[11] J. E. L. E. Gallego, «Universidad Politécnica de Madrid,» [En línea]. Available: https://web.dit.upm.es/~jlopez/publicaciones/mundointernet04.pdf.

[12] P. Security, «Panda Security,» 2017. [En línea]. Available: http://www.pandasecurity.com/spain/support/card?id=31463.

[13] J. C. Snader, VPNs Illustrated: Tunnels, VPNs, and IPsec, Addison-Wesley Profesional, 2005.

[14] Carlos-vialfa, «CCM,» [En línea]. Available: http://es.ccm.net/contents/258-vpn-redes-privadas-virtuales.

[15] J. J. Tomás, «Servicio VPN de acceso remoto basado en SSL mediante openVPN,» Octubre 2008. [En línea]. Available: http://www.ticarte.com/sites/su/users/7/arch/openvpn_con_ssl.pdf.

[16] «omicrono,» [En línea]. Available: http://omicrono.elespanol.com/2012/12/te-explicamos-todo-sobre-x86-y-arm-diferencias-ventajas-y-para-que-sirve-cada-uno/.

[17] F. Doutel, «Xataka,» 18 Agosto 2015. [En línea]. Available: https://www.xataka.com/especiales/guia-del-arduinomaniaco-todo-lo-que-necesitas-saber-sobre-arduino.

[18] «OSMC,» [En línea]. Available: https://osmc.tv/.

[19] «Openelec,» [En línea]. Available: http://openelec.tv/.

[20] M. Á. Navas, «Profesional Review,» 6 Marzo 2016. [En línea]. Available: https://www.profesionalreview.com/2016/03/06/modelo-raspberry-pi-me-compro/.

[21] «Raspberry Shop,» [En línea]. Available: https://www.raspberryshop.es/que-es-raspberry-pi.php.

[22] I. PE, «ComoHacer.eu,» 26 Mayo 2015. [En línea]. Available: https://comohacer.eu/gpio-raspberry-pi/.

[23] S. Crespo, «Alteageek,» [En línea]. Available: https://alteageek.com/como-proteger-tu-servidor-web-con-fail2ban/.

[24] F. Doutel, «Xataka,» 22 Noviembre 2012. [En línea]. Available: https://www.xataka.com/componentes/conoce-a-la-placa-que-quiere-revolucionar-tu-mundo-digital-raspberry-pi-a-fondo-parte-iii.

[25] «Rufus,» [En línea]. Available: https://rufus.akeo.ie/?locale.

[26] «Etcher,» [En línea]. Available: https://etcher.io/.

[27] C. J. W. D. K. H. M. W. Q. S. Narayan, Network Performance Comparison of VPN Protocols, Auckland, New Zealand, 2015.

[28] A. K. S. Khanvilkar, Experimental evaluations of Open-Source Linuxbased, University of Illinois at Chicago, 2004.

[29] E. Jodoin, «Github,» Abril 2014. [En línea]. Available: https://gist.github.com/laurenorsini/10013430.

[30] «No-IP,» [En línea]. Available: https://www.noip.com.

[31] I. Ramírez, «Xataka,» 20 Marzo 2018. [En línea]. Available: https://www.xataka.com/seguridad/que-es-una-conexion-vpn-para-que-sirve-y- que-ventajas-tiene.

[32] O.M.Aguado, «Con más futuro,» 9 Abril 2014. [En línea]. Available: http://www.conmasfuturo.es/principales-diferencias-entre-raspberry-pi-y- arduino-2/..

[33] «OpenVPN,» [En línea]. Available: https://openvpn.net/.

[34] «Snort,» [En línea]. Available: http://www.snort.org.

[35] «Raspberry Pi,» [En línea]. Available: https://www.raspberrypi.org/documentation/configuration/raspi-config.md..

[36] «Raspberry para torpes,» [En línea]. Available: https://raspberryparatorpes.net/instalacion/poner-la-direccin-ip-fija-en-raspbian/#more-2199.

[37] «Putty,» [En línea]. Available: http://www.putty.org/.

[38] «Raspberry Pi,» [En línea]. Available: https://www.raspberrypi.org/documentation/remote-access/ssh/unix.md.

[39] «Snort,» [En línea]. Available: https://www.snort.org/faq/readme-filters.

10. Anexos

Anexo A: Configuración y puesta en marcha de la Raspberry Pi

Una vez se tenga el sistema operativo instalado en una memoria SD o microSD y conectada la placa al monitor con el cable HDMI o mediante SSH o Secure Shell, que por defecto no viene activado y hay que habilitarlo como se explica en la sección, se debe conectar la Raspberry Pi a la corriente utilizando la fuente de alimentación de 2.5 A que se comentaba anteriormente.

Una vez se tenga la plataforma arrancada, en el caso de elegir NOOBS como sistema operativo, se muestra el menú principal de NOOBS donde se ha de seleccionar en la lista el sistema operativo que se quiera instalar, en este caso Raspbian que es el sistema operativo recomendado, esperar a que se descargue y se instale automáticamente.

A.1 Configuración de aspectos iniciales

Una vez se termine de descargar el sistema operativo Raspbian, lo recomendable es realizar una configuración inicial en función de los gustos del usuario. Se puede comenzar con la configuración escribiendo el comando sudo raspi-config en la terminal y aparecerán una serie de opciones de configuración [35].

1. Cambiar la contraseña del usuario (Change User Password): El usuario y contraseña por defecto es “pi” y “raspberry”, es recomendable cambiar la contraseña ya que puede ser un gran agujero de seguridad.
2. Nombre del host (Hostname): Permite modificar el nombre visible en la red de la Raspberry Pi.
3. Opciones de arranque (Boot Options): Permite configurar opciones de arranque.
    a. Desktop / CLI: Permite cambiar si al arrancar el sistema operativo se carga el escritorio, que es la configuración por defecto, o si solo se carga la terminal. También permite configurar el autologin o por el contrario se pedirá al iniciarse la placa introducir el usuario y contraseña.
    b. Wait for Network at Boot: Si se habilita esta opción la placa espera a estar conectada a la red (Wi-Fi, Ethernet) para iniciar algunos servicios.
    c. Splash Screen: Permite configurar si se quiere mostrar una pantalla de bienvenida al usuario.

4. Opciones de localización (Localisation options): En esta opción se puede configurar el lenguaje del sistema operativo, la zona horaria y la distribución del teclado.
    a. Establecer el idioma para el sistema operativo, por defecto viene seleccionado “en_GB”, y se ha añadido “es_ES UTF-8”.
    b. Permite establecer el horario local, se ha seleccionado Europe, Madrid.
    c. Configurar el teclado, lo normal es seleccionar el tipo de teclado “Generic 105-key (Intl) PC”, el idioma por defecto está en inglés británico y se ha cambiado a “Spanish”. La ubicación de la tecla “AtlGr” se ha seleccionado “The default for the keyboard layout”, que es lo normal. A continuación, solicita la tecla para componer “AltGr” y se ha seleccionado “No compose key”. Por último, en esta opción, pregunta si se desea usar la combinación de teclas “Ctrl + Alt + Borrar” para salir del modo de interfaz gráfica, se ha seleccionado la opción por defecto que es No.
    d. Permite seleccionar el país en el cual se va a usar la Raspberry Pi, se ha seleccionado Spain.
5. Opciones de interconexión (Interfacing Options): Permite configurar la conexión con los periféricos.
    a. Camera: Si se habilita esta opción se activa el puerto para permitir la comunicación entre el controlador de la cámara y la CPU.
    b. SSH: Habilitar / deshabilitar el protocolo SSH que permite acceder de forma remota a la consola de comandos de la Raspberry Pi desde otro ordenador, por defecto está deshabilitado. Se ha habilitado esta opción y para que se pueda realizar la conexión se debe conectar la placa al router o mediante Wi-Fi.
    c. VNC: Habilitar / deshabilitar la opción VNC que permite controlar de forma remota la interfaz de escritorio de la Raspberry Pi desde otro ordenador o dispositivo móvil ejecutando VNC Server.
    d. SPI: Habilitar / deshabilitar interfaces SPI y cargar automáticamente el módulo SPI kernel, necesario para productos como PiFace.
    e. I2C: Permite habilitar / deshabilitar interfaces I2C y cargar automáticamente el módulo del kernel I2C.

f. Serial: Habilitar / deshabilitar los mensajes de shell y kernel en la conexión serie.
g. 1-Wire: Permite habilitar / deshabilitar la interfaz one-wire.
h. Remote GPIO: Permite habilitar / deshabilitar el acceso al servidor GPIO a través de la red.

6. Overclock: La placa viene configurada por defecto para funcionar a 700 MHz, pero este valor se puede aumentar para sacar más partido al rendimiento de la misma. La fundación Raspberry Pi garantiza que la placa funcionará hasta a 1 GHz en su CPU y hasta 600 MHz su memoria RAM sin ningún problema de rendimiento.
7. Opciones avanzadas (Advanced Options): Permite configurar las opciones avanzadas.
    a. Expand Filesystem: Esta opción permite expandir el sistema operativo para que se utilice todo el espacio disponible en la tarjeta. No es necesario ejecutar esta opción si el SO se ha instalado con NOOBS.
    b. Overscan: En función del tamaño de la pantalla del monitor que se vaya a utilizar, es necesario habilitar o deshabilitar esta opción para que la imagen se ajuste a la pantalla y se corrija la resolución.
    c. Memory Split: Permite modificar la cantidad de memoria disponible para la GPU.
    d. Audio: Permite forzar la salida de audio a través de HDMI o un conector de 3,5 mm.
    e. Resolution: Permite establecer un modo de resolución determinado.
    f. GL Driver: Permite la aceleración gráfica de la pi, al ser una función experimental viene desactivada por defecto.
8. Update: Permite actualizar Raspbian y los paquetes que vienen ya en él, para ello la Raspberry pi debe estar conectada por Ethernet.
9. About raspi-config: Información sobre la configuración inicial de la placa.

Tras esta configuración será necesario reiniciar la placa: sudo reboot

A.2. Conexión vía SSH a la Raspberry Pi

Tras haber realizado la configuración inicial, se ha establecido una IP fija para la placa para poder conectarse más rápidamente por SSH (Secure Shell) siendo esta una herramienta que permite acceder a máquinas remotas a través de la red y controlarlas

mediante el intérprete de comandos. El requisito indispensable para poder realizar la conexión vía SSH es tener habilitada esta opción como se ha comentado en el anexo.

A.2.1. Configuración de aspectos iniciales.

Por defecto cuando se conecta por cable, la Raspberry Pi obtiene la dirección con la que se identifica en la red por DHCP (Dynamic Host Configuration Protocol) que es asignada por el router que le asocia una IP que no está siendo usada. El router suele guardar esta asociación entre IP y equipo, pero puede darse que se pierda esta asociación y al reiniciar la placa se le asocie una IP diferente. La solución es tener siempre la misma IP, para ello hay que fijarla manualmente en la configuración de la red [36].

Para ello, hay que introducir el siguiente comando en la terminal de la Raspberry Pi:

```
sudo nano /etc/network/interfases
```

Una vez se esté dentro de este fichero se ha de modificar para establecer una IP fija. Para saber que IP fija establecer, lo recomendable es ver la configuración del router para saber si existe un rango reservado y si no seleccionar una IP alta (el máximo es 254). El parámetro nestmask se refiere a la máscara de red y el término gateway a la IP del router, que por defecto son siempre los mismos valores. En la *Figura A.1* se muestra cómo quedaría el fichero tras la modificación.

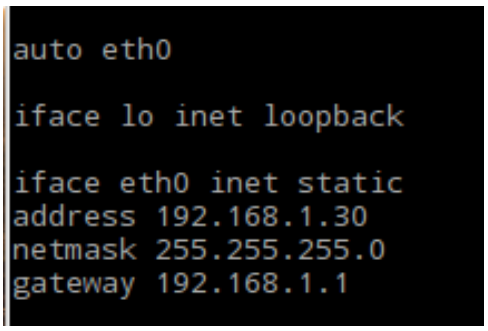


Figura A.1. Configuración del fichero interfaces.

Una vez se ha guardado el fichero con los cambios, se debe reiniciar la Raspberry pi para que se apliquen los cambios con el comando sudo reboot.

Cuando se vuelva a iniciar la placa, con el comando ifconfig se comprueba que la IP se ha modificado y es siempre la misma (*Figura A.2*).

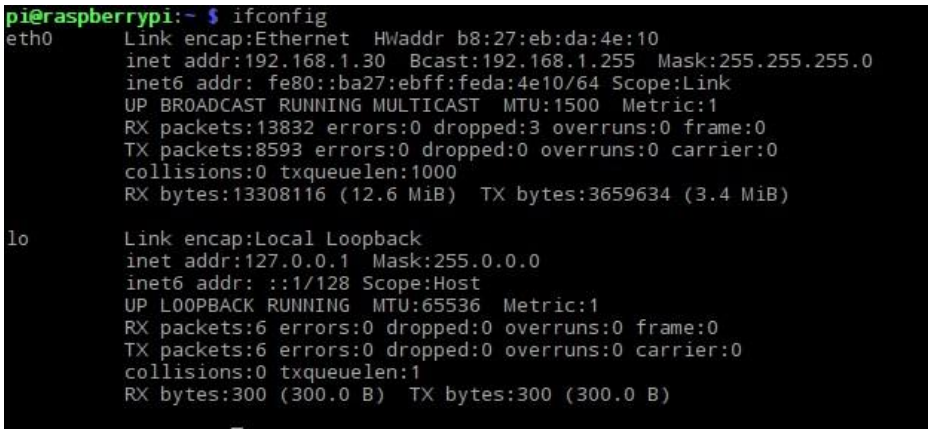


Figura A.2. IP fija de la Raspberry Pi.

Si se quiere asignar la IP fija en lugar de por cable por Wi-Fi, es la misma solución a diferencia que en el fichero interfaces con el que se ha accedido con el comando sudo nano /etc/network/interfaces, habría que añadir este texto:

```
auto  wlanX
iface  wlanX  inet  static
address  192.168.1.XX
netmask  255.255.255.0
gateway  192.168.1.1
wpa-ssid  name_network_wifi
wpa-psk  pass_network_wifi
```

Donde la X de wlan, corresponde al número asociado a wlanX que se muestra a cada usuario en la terminal de Raspberry Pi cuando ejecuta el comando ifconfig. Y XX de address corresponde a la IP fija elegida con los criterios ya comentados.

Tras haber establecido una IP fija, en Windows desde el programa PuttY [37] se puede acceder de forma remota introduciendo la IP y aceptando un certificado que aparece la primera vez que se realiza la conexión. Una vez se cargue la conexión, se pedirá el nombre de login, normalmente “pi”, y la contraseña establecida por el usuario. Si la conexión se realiza desde Mac o Linux [38], habrá que utilizar el comando ssh -x pi@<IP_PLACA>, donde en este caso habría que sustituir <IP_PLACA> por 192.168.1.30, quedando finalmente el comando ssh -x pi@192.168.1.30.

Anexo B: Generar nombre de dominio dinámico.

Si no se cuenta con una IP publica estática, se puede usar un sistema de DNS dinámico que permite configurar un nombre de dominio y asignarlo a la IP pública, permitiendo acceder a nuestra IP pública mediante el nombre de dominio, aunque la IP pública cambie. Se va a explicar cómo crear un host con el servicio No-IP en la página https://www.noip.com.

1. Lo primero que hay que hacer, si no se está registrado en la página, es registrarse en la web oficial, en la opción Sign Up (*Figura B.1*).

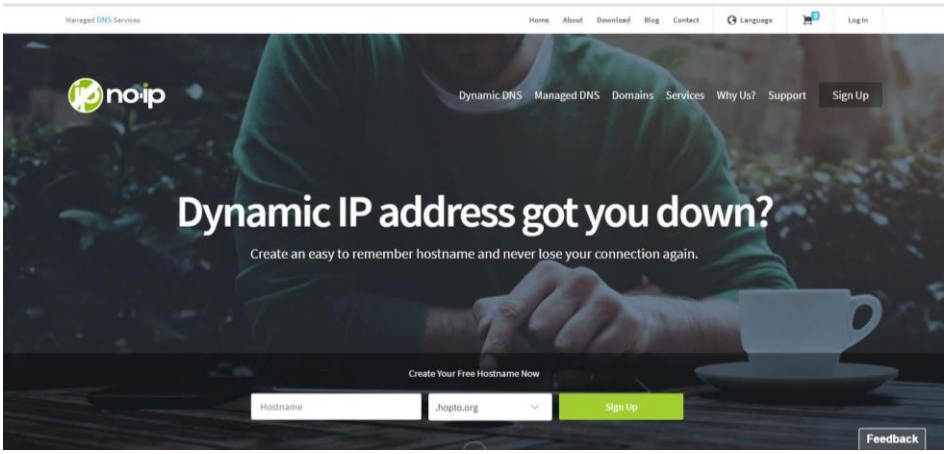


Figura B.1: Sign Up en No-IP.

2. Una vez se esté en esta pestaña, permite crear una cuenta gratuita y pide unos campos a rellenar, el email, un nombre de usuario, contraseña y el nombre que se le va a dar al host.
3. Si no se ha iniciado sesión directamente al registrarse, iniciar sesión en la opción Sign In con el correo electrónico o nombre de usuario y contraseña elegidos.
4. Iniciado sesión, se accede al panel de control de la página web donde se puede crear y administrar un máximo de tres hosts de forma gratuita durante 30 días. Para ver los hosts que se tienen creados, seleccionar en el panel de control Dynamic DNS / Hostnames como se muestra en la *Figura B.2*.

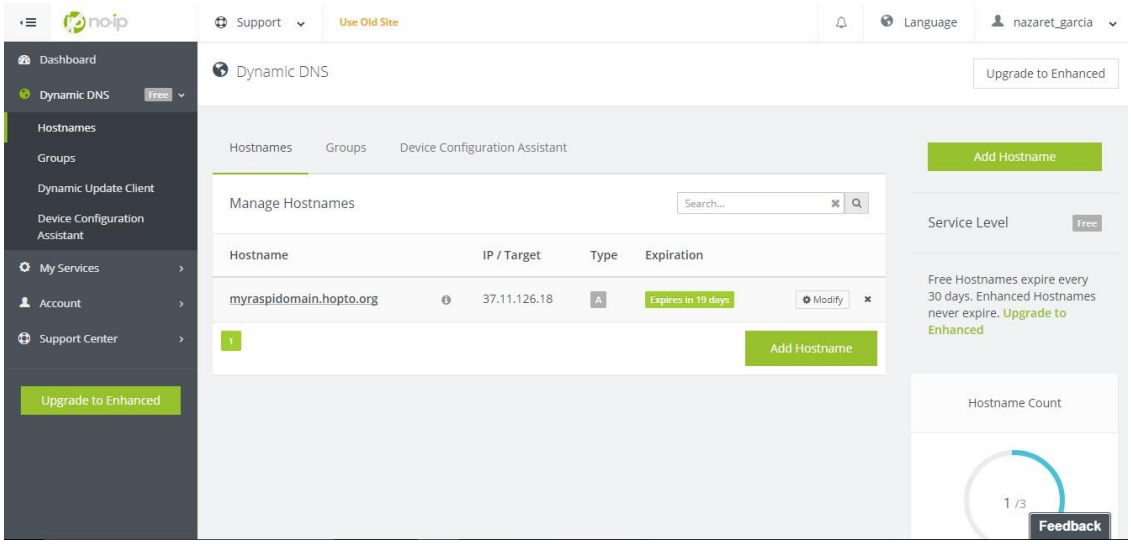


Figura B.2: Panel de control del usuario en No-IP.

5. Para crear un host, seleccionar Add Hostname, dar un nombre de hostname, seleccionar un dominio gratuito y la dirección IP asignada es la que se debe introducir en el fichero Default.txt creado en el Capítulo 4 en la sección *4.2.2. Lado del cliente*. Una vez rellenado estos datos, pulsar en Create host y ya estaría creado (*Figura B.3*).

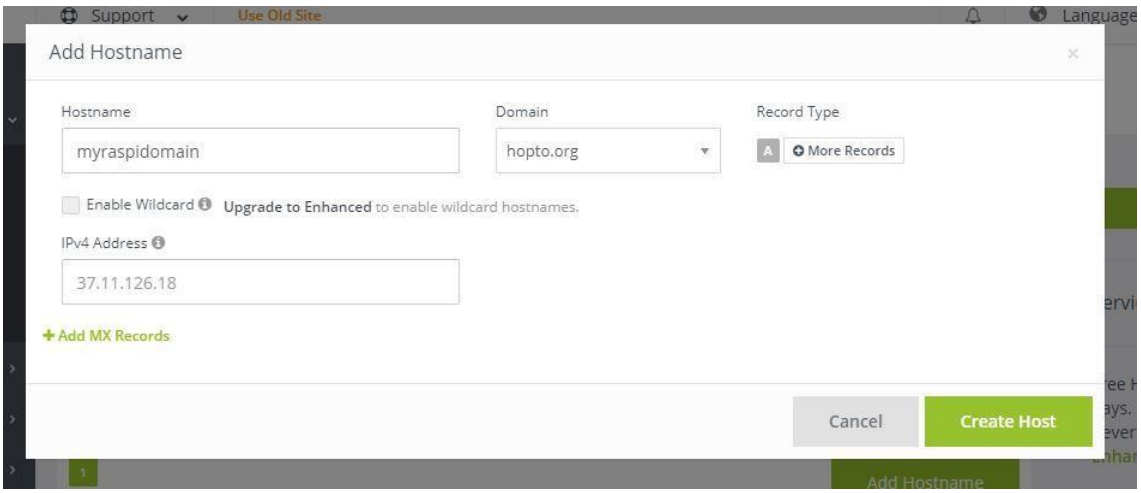


Figura B.3: Creación de host en No-IP.

Una vez registrado el host, hay que configurar en la Raspberry Pi los datos de la cuenta No-IP. Para ello, hay que descargar el cliente DUC (*Dynamic Update Client*) con los siguientes comandos.

```
cd  /usr/local/src/
wget  http://www.no-ip.com/client/linux/noip-duc-
linux.tar.gz
tar  xf  noip-duc-linux.tar.gz
cd  noip-2.1.9-1/
```

```
make  install  (Si falla el comando make, instalar el compilador gcc
con el comando sudo  apt-get  install  build-essential)
```

6. Al realizar la instalación del DUC se solicitan los datos de la cuenta No-IP, el nombre de usuario (nazaret_garcia en mi caso) y la contraseña.
7. Como se ha hecho en otras ocasiones, hay que crear un fichero para que el servicio No-IP se inicie si se reinicia la Raspberry Pi:

```
sudo  nano  /etc/init.d/noip2
```

Con el siguiente contenido:

```
#!  /bin/bash
###  BEGIN  INIT  INFO
###  END  INIT  INFO
sudo  /usr/local/bin/noip2
```

Y se le dan permisos de ejecución al fichero:

```
sudo  chmod  +x  /etc/init.d/noip2
```

Y para terminar con esta configuración utilizamos el comando update-rc.d que automatiza el proceso de creación y borrado de enlaces a los scripts de inicio, con el fin de iniciar/parar servicio:

```
sudo  update-rc.d  noip2  defaults
```

Anexo C: Conexión con la VPN.

Una vez generado como mínimo un usuario, se pueden instalar clientes en función del sistema operativo. Para conectarse con la VPN en Windows, existe el cliente openVPN GUI.

Lo primero que se debe hacer es descargar e instalar el cliente para Windows con las opciones predeterminadas.

Cuando se haya terminado la instalación, hay que copiar el archivo de cada cliente con extensión .ovpn en la carpeta C:\Program Files\OpenVPN\config. Como el archivo ha sido creado con el programa openvpn no hace falta añadirle la ruta de los certificados, la dirección del servidor OpenVPN y redireccionar el tráfico porque ya está correctamente configurado.

Tras haber hecho lo anterior, ya está hecha la parte de configuración, a continuación, se ejecuta el programa y notifica que el cliente ya está inicializado como se muestra en la *Figura C.1*. Una vez inicializado el cliente, se muestra en los iconos ocultos de la barra de tareas, hacer clic derecho en el icono de OpenVPN GUI y pulsar en Connect (*Figura C.2*).

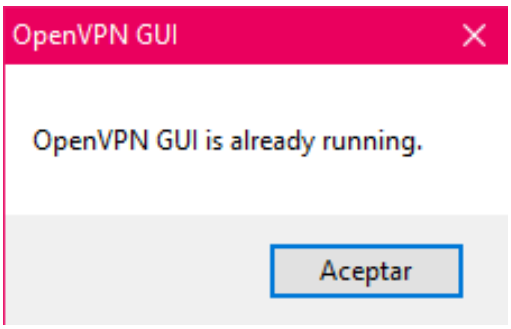


Figura C.1: Cuadro de diálogo OpenVPN GUI.

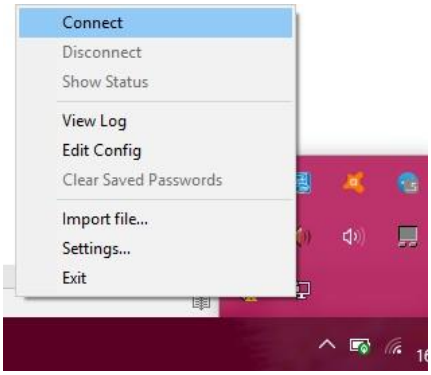


Figura C.2: Conexión OpenVPN GUI.

Con la conexión establecida, se podrá navegar por Internet y el tráfico pasará por el túnel. Se puede comprobar la IP pública en cualquier página de Internet y comprobar que la IP que ve el exterior es la del host.