



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Servidor multimedia con VPN en Linux

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Cobo Canela, Alfonso

Tutor/a: Pons Terol, Julio

CURSO ACADÉMICO: 2021/2022

Agradecimientos

Quiero agradecer su ayuda y apoyo a mi tutor Julio Pons, a mi compañero Guillermo Ruíz ya que sin ellos este trabajo no podría haberse completado, también a mis padres Ana María y Alfonso por su ayuda y esfuerzo, y por último a Irene Romero por todo el apoyo y tiempo que me ha dedicado.

Índice general

Índice general	V
Índice de figuras	VII

Resumen	IX
Abstract	XI
Resum	XIII
1 Introducción	1
1.1 Motivación	1
1.2 Objetivos	1
1.3 Impacto esperado	2
2 Contexto tecnológico	3
2.1 Crítica al contexto tecnológico	4
2.2 Propuesta	4
3 Análisis del problema	5
3.1 Análisis de la seguridad y protección de datos	5
3.2 Análisis energético	5
3.3 Identificación y análisis de soluciones posibles	6
3.4 Solución propuesta	6
3.5 Presupuesto	6
4 Diseño de la solución	9
4.1 Arquitectura del sistema	9
4.2 Diseño detallado	10
5 Tecnología utilizada	13
5.1 Sistema Operativo	13
5.1.1 SO para servidor VPN	13
5.1.2 SO para el servidor multimedia	14
5.2 SSH	14
5.3 SMB	15
5.4 Comparativa VPNs	15
5.4.1 PPTP	16
5.4.2 Ipsec/L2TP	16
5.4.3 OpenVPN	17
5.4.4 Ipsec/IKEv2	18
5.5 Reproductor multimedia	19
5.5.1 Emby	20
5.6 Panel de control	21
5.7 DDNS	21
5.8 Wake on LAN	22
6 Desarrollo de la solución propuesta	23
6.1 Montaje de los ordenadores	24
6.1.1 Instalación del ventilador para la Raspberry PI	24

6.2	Instalación sistema operativo	25
6.2.1	Raspberry PI OS modo «headless»	25
6.3	Instalación software	26
6.3.1	VPN	26
6.3.2	SMB	28
6.3.3	SSH	28
6.3.4	Webmin	29
6.3.5	Servidor Multimedia	30
6.4	Configuración Wake On Lan	31
6.5	Creación de scripts	31
6.6	Configuración del rúter	32
6.7	Configuración del DDNS	33
7	Seguridad	35
7.1	VPN	35
7.1.1	Problema seguridad IKE (modp1024)	35
7.1.2	Solución Windows 10	36
7.2	Firewall	36
7.3	Fail2ban	37
7.4	Permisos UNIX, SAMBA y SSH	37
8	Pruebas	39
8.1	Rendimiento LAN	39
8.2	Rendimiento VPN	41
8.3	Funcionamiento Scripts	42
8.4	Cifrado	43
9	Limitaciones y posibles mejoras	45
9.1	Multithreading	45
9.1.1	Posible solución	45
9.2	Ancho de banda de internet	46
10	Conclusiones	49
11	Relación del trabajo desarrollado con los estudios cursados	51
	Bibliografía	53
<hr/>		
Apéndices		
A	Configuración del sistema	59
A.1	Dirección IP estática	59
A.2	Crea, eliminar y modificar usuarios	59
A.3	Instalación y configuración de la VPN	60
A.4	Instalación del paquete SAMBA	61
A.5	Instalación SSH y configuración	62
A.6	Scripts	64
B	Configuración de elementos de seguridad	65
B.1	Reglas IPtables	65
C	Objetivos para el Desarrollo Sostenibles (ODS)	67

Índice de figuras

3.1	Factura del PC que se será el servidor de NAS	7
4.1	Arquitectura de la red	9
5.1	Gráfico rendimiento OpenVPN con distintos cifrados[16]	17
5.2	Datagramas IPsec en donde se aplican los distintos protocolos [21]	19
5.3	Máquina Virtual de «Oracle Cloud»	22
6.1	Diagrama de relación entre procesos del sistema	23
6.2	Tabla de pines y voltajes del GPIO[31]	24
6.3	Ventilador y cable de adaptación	25
6.4	Archivo SSH creado en el directorio raíz de la tarjeta SD.	25
6.5	Interfaz del rúter donde se muestra el «hostname» y la IPv4 otorgada.	26
6.6	Diagrama de las dependencias entre certificados y llaves.	26
6.7	Diagrama de mensajes entre cliente y servidor para la autenticación de la conexión SSH [37]	29
6.8	Ejemplo de estructura de directorios en la partición multimedia de SMB	30
6.9	Interfaz Emby para seleccionar que tipo de contenido tendrá la carpeta	30
6.10	Interfaz de «BIOS» en la que se ve la opción para despertar el servidor mediante dispositivos conectados a la interfaz PCIe	31
6.11	Web propia creada en PHP donde se ven los dos botones para encender o apagar el ordenador	32
6.12	Captura de la configuración del «Port Forwarding» del puerto 4500	33
6.13	Captura del apartado «Dynamic DNS» de la API de IONOS	34
7.1	Captura del registro modificado	36
8.1	Gráfica que representa la velocidad media de transferencia de un archivo variando su tamaño en un disco HDD	39
8.2	Captura del resultado de transferir 1000 archivos de 1KiB sin comprimir	40
8.3	Gráfica que representa la velocidad media de transferencia de un archivo variando su tamaño en un disco SSD	40
8.4	Gráfica que representa la velocidad media de transferencia de un archivo variando su tamaño con la VPN	41
8.5	Registro donde indica el algoritmo seleccionado por el usuario	41
8.6	Test de dos usuarios concurrentes usando la VPN donde se ve que solo se está usando un núcleo de la CPU del servidor VPN.	42
8.7	Registros donde se indica que la creación del IKE-SA ha sido correcta, esto indica que el usuario conectado ha superado el EAP y tiene el certificado correcto.	42
8.8	Registros donde se indica que el «sniffer» ha detectado correctamente que la creación del IKE-SA ha sido correcta. La información censurada son direcciones MAC y el nombre del usuario.	42
8.9	Datagrama encapsulado en UDP de un paquete ICMP cifrado.	43

8.10	Paquete ICMP sin cifrar en el que se ve su contenido.	43
9.1	Diagrama de procesos Docker y proxy inverso.	46
9.2	Diagrama de conexión entre el servidor VPN y un usuario en el que se ve que la velocidad de subida de la VPN es la velocidad de bajada del usuario .	46
A.1	Huella gráfica del par de llaves RSA de ejemplo	63

Resumen

Se ha diseñado un sistema híbrido formado por una «Raspberry PI 4B» que actúa como servidor VPN y un ordenador de sobremesa que se ha configurado como servidor NAS y multimedia.

Este diseño surge para solucionar el cuello de botella que se origina en la «Raspberry PI 4B» al transferir grandes cantidades de datos a altas velocidades, en caso de que la «Raspberry PI 4B» actuara como servidor NAS, es por esto que se plantea el uso de un ordenador más potente. Además para no aumentar el gasto energético innecesariamente se ha diseñado un software que permite desde la Raspberry PI encender y apagar el ordenador tanto de forma automática, dependiendo de la VPN, como de forma manual.

Se han instalado y configurado todos los elementos necesarios en ambos equipos para dotar el sistema de una seguridad adecuada y se han hecho pruebas del rendimiento de la NAS y de la red local dando como resultado que el cuello de botella es, en ambos casos, la velocidad de transmisión de las redes.

Palabras clave: híbrido, Raspberry PI, VPN, SMB, multimedia, Emby, software, cuello de botella, red local

Abstract

A hybrid system has been designed consisting of a «Raspberry PI 4B» that acts as a VPN server and a desktop computer that has been configured as a NAS server and multimedia server.

This design arises to solve the bottleneck that originates in the «Raspberry PI 4B» when transferring large amounts of data at high speeds, in case the "Raspberry PI 4B" acted as a NAS server, which is why the use of a more powerful computer is proposed. Moreover, in order to not increase energy consumption unnecessarily, a software has been designed that allows the computer to be turned on and off from the Raspberry PI both automatically, depending on the "VPN", and manually.

All the necessary elements have been installed and configured on both computers to provide the system with adequate security and the performance of the "VPN" and the local network have been tested, resulting in the bottleneck being, in both cases, the transmission speed of the networks.

Key words: hybrid, Raspberry PI, VPN, SMB, multimedia, Emby, software, bottleneck, local network

Resum

S'ha dissenyat un sistema híbrid format per una «Raspberry PI 4B» que actua com a servidor VPN i un ordinador de sobretaula que s'ha configurat com a servidor NAS i servidor multimèdia.

Aquest disseny sorgix per a solucionar el coll de botella que s'origina en la «Raspberry PI 4B» al transferir grans quantitats de dades a altes velocitats, és per açò que es planteja l'ús d'un ordinador més potent. A més per a no augmentar el gasto energètic innecessàriament s'ha dissenyat un programa que permet des de la Raspberry PI encendre i apagar l'ordinador tant de forma automàtica, depenent de la VPN, com de forma manual.

S'han instal·lat i configurat tots els elements necessaris en els dos equips per a dotar el sistema d'una seguretat adequada i s'han fet proves del rendiment de la VPN i de la xarxa local donant com resultat que el coll de botella és, en els dos casos, la velocitat de transmissió de les xarxes.

Paraules clau: híbrid, Raspberry PI, VPN, SMB, multimèdia, Emby, programa, coll de botella, xarxa local

CAPÍTULO 1

Introducción

1.1 Motivación

La idea original de este proyecto surgió por la necesidad de poder acceder a archivos en un lugar donde no se aceptan dispositivos de almacenamiento externos. En un principio se usó Drive¹, la herramienta de almacenamiento en la nube de Google, pero los 15GB que ofrece Google en la versión gratuita no son suficientes. Después se probó con MEGA², un servicio parecido a Drive, el cual ofrece 50GB en su versión gratuita pero tiene límite de subida y descarga de datos. Es por estas limitaciones que se decidió crear un servidor de archivos en red o NAS («Network Attached Storage») con un ordenador que no se utilizaba.

Una de las formas que permite acceder desde cualquier lugar al servidor NAS es mediante una red privada virtual o VPN («Virtual Private Network») que permite crear una conexión segura cuando se estén utilizando redes que no sean confiables.

Posteriormente para aprovechar aún más las capacidades del ordenador se decidió instalar un servidor multimedia para poder acceder a contenido multimedia y reproducirlo desde cualquier lugar.

A raíz de la reciente subida del precio de la luz se propuso el NAS híbrido con una Raspberry PI, ya que la mayor parte del tiempo el NAS se quedaba encendido sin ser utilizado. Con esta nueva versión híbrida se aprovecha la potencia de un procesador «x86» y la eficiencia de un procesador de arquitectura ARM (Advanced RISC Machine), ya que cuando se necesite la potencia del NAS la Raspberry PI encenderá el servidor de NAS.

1.2 Objetivos

El principal objetivo es crear un servidor VPN y un servidor multimedia/NAS que sea eficiente y seguro. Este sistema servirá principalmente para poder consumir contenido multimedia desde cualquier lugar y tener una conexión segura cuando, por ejemplo, se usen conexiones WiFi públicas. Además, también se utilizará el protocolo en red SMB (Server Message Block) para poder compartir archivos o dispositivos dentro de la red local que, con la ayuda de la VPN, serán accesibles desde fuera de esta red.

Por otro lado se utilizará una opción de servicio de DNS (Domain Name Server) dinámico DDNS (Dynamic Domain Name Server) para poder tener un dominio fijo independientemente de que la dirección IP cambie.

¹https://www.google.com/intl/es_es/drive/

²<https://mega.io/>

Se hará un estudio sobre diversos paneles de control y se instalará la mejor opción en cada servidor para poder administrarlos y controlarlos de manera más sencilla y rápida.

Por último se desarrollará un conjunto de «scripts» y páginas web para que el servidor VPN, en este caso la Raspberry PI, pueda encender y apagar el servidor multimedia/NAS ahorrando en energía cuando este último no se use.

1.3 Impacto esperado

Se espera que este sistema ofrezca mejores prestaciones que otros servicios como Drive, Mega o Dropbox³ por un menor coste.

Por una parte el espacio de almacenamiento aumentaría, no se tendría un límite de ancho de banda ni tampoco de la cantidad de datos a enviar o descargar, además se dispone de conexión cifrada para cuando se esté en conexiones poco seguras, por ejemplo «WiFis» públicos. Por otra parte el gasto energético del sistema disminuye y también la contaminación acústica, sobretodo si el servidor está en habitaciones de convivencia.

Además este trabajo también cumple principalmente con dos objetivos de desarrollo sostenible (ODS), que son «Producción y consumo responsable», ya que el gasto energético del servidor solo se produce a la hora de su uso y «Acción por el clima» porque el consumo energético total se reduce y por ende la huella de carbono de este sistema reduciendo así las emisiones de gases de efecto invernadero, ver apéndice sobre los «ODS» C.

³<https://www.dropbox.com/es>

CAPÍTULO 2

Contexto tecnológico

El uso actual de la nube en Europa está todavía en fase de desarrollo, ya que solo el 40 % de las empresas europeas usan esta tecnología siendo el 60 % restante servidores propios «On Premise». Igualmente la nube será la tecnología más relevante en el futuro ya que entre 2020 y 2021 el uso de esta tecnología en las empresas se ha mantenido el mismo o ha incrementado [1]. Con esto se intenta explicar que ya existen tecnologías de almacenamiento en la nube, tales como: Drive, MEGA o Cloudflare. Incluso se pueden alquilar servidores en la nube para poder montar ahí el NAS.

Otra tecnología que tiene mucho recorrido es la VPN, entre todos los protocolos que existen destacamos los 4 más usados [2]: PPTP, «L2TP/IPsec», «IKEv2/IPsec» y «OpenVPN». Aún así los protocolos de VPN no son los únicos que sirven para conectar redes, también encontramos el protocolo SSH («Secure Shell») que a través de intercambios de llaves y redirección de puertos puede crear conexiones entre redes aunque de forma más compleja. Además existen otros protocolos de terceros como Citrix¹ o TeamViewer² que sirven para cumplir ciertas funciones de la VPN sin la necesidad de conectar las redes.

Por otro lado hay empresas que ofrecen sus servicios de VPN de forma gratuita como: IPVanish o PrivateVPN.

Para poder consumir contenido multimedia también existen muchas opciones: Spotify³, YouTube⁴, Netflix⁵ pero para consumir contenido propio es necesario otro software. En este caso servidores de «streaming», entre los que encontramos: Plex⁶, Kodi⁷ o Emby⁸.

Obviamente el uso de una Raspberry PI como servidor de WOL («Wake On LAN») no es algo original de este trabajo. Existen multitud de artículos sobre como utilizar este tipo de servidores, por ejemplo, como se muestra en [3] donde se explica como utilizar este tipo de servidores para encender dispositivos del entorno. Hay diversos software que realizan la misma función como «etherwake» o «wakeonlan» para Linux.

¹<https://www.citrix.com/es-es/>

²<https://www.teamviewer.com/es/>

³<https://open.spotify.com/>

⁴<https://www.youtube.com/>

⁵<https://www.netflix.com/>

⁶<https://www.plex.tv/>

⁷<https://kodi.tv/>

⁸<https://emby.media/>

2.1 Crítica al contexto tecnológico

El uso de almacenamiento en la nube, VPN de terceros o contenido multimedia trae consigo una confianza en la empresa que ofrece el servicio, porque en el momento en el que se envían los datos se ha perdido el control sobre ellos.

Se han creado trabajos similares a este los cuales hablan sobre las tecnologías que se van a usar.

En el primer trabajo «Configuración de un servidor VPN alojado en una «Raspberry» [4] no se habla sobre el rendimiento de VPN ni como la Raspberry PI puede ser un punto de cuello de botella en cuanto al rendimiento de la VPN o SMB. Además que, según mi punto de vista, no aprovecha el potencial de la Raspberry PI porque se trata como un ordenador «normal» cuando su arquitectura es completamente distinta.

Por otro lado el segundo trabajo «Implementación de una VPN (Virtual Private Network) usando el estándar IPsec» [5] profundiza mucho más la tecnología IPsec (Internet Protocol Security) y la creación de la VPN, pero se explica a través del contexto tecnológico de 2001. Esto es problema ya que en 21 años: se han creado nuevos protocolos de VPN, han cambiado las características de la red, se ha aumentado las prestaciones, etc. Por eso es necesario expandir este tema y actualizarlo.

2.2 Propuesta

Este proyecto se parece a trabajos anteriores en cuanto al uso de una Raspberry PI como servidor VPN y el uso de IPsec como principal protocolo de seguridad para la VPN. Aún así se hará un estudio sobre los protocolos actuales a utilizar y su comparativa con protocolos similares. Además se realizarán pruebas para asegurar el mejor rendimiento y detectar posibles cuellos de botella en el sistema. Por otra parte también se centrará en el apartado de eficiencia, es decir obtener las mejores prestaciones consumiendo lo mínimo posible combinando Raspberry PI y un ordenador de sobremesa.

CAPÍTULO 3

Análisis del problema

3.1 Análisis de la seguridad y protección de datos

En la actualidad los casos de ataques a través de la red han aumentado drásticamente en España [6] esto se traduce en una desconfianza a la hora de transferir datos o subirlos a la nube. Además ninguna herramienta que contratemos a terceros garantiza seguridad como bien demuestra la página web haveibeenpwned.com¹, la cual es una recopilación de más de 11500 millones de cuentas y contraseñas que han sido filtradas. Entre las empresas que sufrieron las brechas más grandes encontramos: 000webhost, Adobe, Avast, Audi, CD Projekt RED, Comcast, Domino's, Dropbox, etc. Como se ve ni las empresas más grandes del mercado están exentas de sufrir ataques y filtrar información privada de sus clientes.

Si además añadimos que muchas de estas empresas guardan y utilizan los datos de sus clientes para su propio beneficio, la privacidad desaparece.

Por otra parte la seguridad no es solo la privacidad de los datos si no también su persistencia, si las empresas no otorgan la seguridad de que no se perderán los datos es necesario encontrar una solución para evitar este problema, un ejemplo fue el incendio que sufrió la empresa de alojamiento «OVH» en sus instalaciones que destruyó los datos de muchos clientes entre ellos particulares y empresas [7]. Todo aquel usuario, afectado por el incendio, que administrase su propio servidor y que no dispusiera en su momento de una copia de sus datos en otro centro de datos, los perdió.

Es por estas razones que un NAS propio solucionaría a estos problemas. Ya que la información sería privada y permitiría que se pudieran hacer copias de seguridad desde cualquier máquina.

3.2 Análisis energético

A causa de la crisis energética actual, muchos europeos se están «apretando los cinturones» para reducir el gasto en electricidad. Es por eso que no sería conveniente tener un ordenador siempre encendido para usar un NAS de vez en cuando. Teniendo en cuenta que un ordenador promedio consume 60W la hora sin soportar ninguna carga y que el precio de la energía en el momento de redactar este trabajo es de 0,31€/kWh esto supondría un gasto de 162,94€ anuales, 13,58€ mensuales.

$$\frac{W * e / kWh * horas * días}{1000}$$

¹<https://haveibeenpwned.com/>

Por tanto, un sistema compuesto de solo un ordenador sería relativamente caro. Por esta razón el uso de una Raspberry PI ayudaría a rebajar este coste energético ya que este ordenador solo consume una media de 6W la hora sin soportar ninguna carga. Aplicando la fórmula anterior se obtiene que los gastos se reducen a 16,29€ anuales.

Hasta el momento no se ha tenido en cuenta la carga de los ordenadores por simplicidad al calcular el gasto energético. En la práctica habría que tener en cuenta diversos factores a la hora de obtener estos gastos pero implicaría realizar cálculos más complejos.

3.3 Identificación y análisis de soluciones posibles

Tal y como se ha dicho en anteriores apartados, existen múltiples soluciones para este problema. Existen soluciones que ya se han implementado como en «Implementación de un servidor VPN en una Raspberry»[4] que en vez de tratar de un sistema de dos máquinas utiliza solamente una, reduciendo así el consumo energético a costa de perder rendimiento.

Otra solución es contratar un servidor en la nube y montar el NAS ahí, aunque esta implementación supondría un coste más elevado y, como hemos hablado en el apartado de «Análisis de la seguridad y protección de datos», depender de los sistemas de una empresa puede acarrear problemas de persistencia de datos y/o privacidad si no se dispone de copias de seguridad.

Igualmente estos sistemas anteriores tienen una ventaja muy importante y es que son más sencillos de configurar y mantener, ya que disponen de una sola máquina.

Para este caso, servidor multimedia y NAS, el rendimiento que ofrece la Raspberry PI es insuficiente pero su eficiencia energética es muy buena y es por esta última razón que, junto a un ordenador más potente, se utilizará en este sistema híbrido. De esta forma se gastará menos dinero pero no perderemos potencia de codificación para transmitir vídeos o para usar la NAS.

3.4 Solución propuesta

La propuesta de este trabajo es hacer un servidor multimedia híbrido entre una Raspberry PI y un ordenador de arquitectura x86 de sobremesa al cual el usuario se pueda conectar mediante VPN. Esta solución conlleva la utilización de tres servidores principales: servidor de VPN, servidor SMB y servidor multimedia.

El servidor de VPN estará alojado en la Raspberry PI y el resto de servidores en el ordenador de sobremesa, al que a partir de ahora se le hará referencia como NAS. La idea principal es ahorrar energía encendiendo el ordenador de sobremesa solo cuando se necesite. Esto se conseguirá analizando los registros del servidor VPN, el cual siempre estará encendido, y cuando haya una conexión exitosa se encenderá el ordenador de sobremesa a través de paquetes WOL enviados desde la Raspberry PI.

3.5 Presupuesto

Como ya hemos comentado antes el sistema que se va a implementar se compone de 2 máquinas: un ordenador de sobremesa de arquitectura x86 64-bit y una «RaspbberyPI 4B» de arquitectura ARM.

El valor de ambas máquinas ronda los 170€ (el ordenador de sobremesa 100€ 3.1 y la Raspberry 70€), aunque últimamente la «RaspberryPI 4B» ha subido mucho de precio por culpa del desabastecimiento de microchips llegando hasta los 200€ en algunas tiendas «online».

Información del pedido		Dirección de envío	Total del pedido	
Comprador	██████████	Alfonso Cobo	1 artículo	107,00 EUR
Vendedor	██████████	██	Envío	11,99 EUR
Tramitada el	20 dic. 2021	VALENCIA, Comunidad Valenciana ██████████	Total del pedido	118,99 EUR
Forma de pago	██████████	España		
Pagado el	20 dic. 2021			

Artículos comprados a ██████████

N.º de pedido: ██████████

Cantidad	Nombre del artículo	Servicio de envío	Precio del artículo
1	HP Workstation Elitedesk 705 G2 SFF AMD A8-8650B 16GB 500GB HDD Schule Klasse A (██████████)	Standard International	107,00 EUR

Figura 3.1: Factura del PC que se será el servidor de NAS

A parte de las máquinas es necesario «hardware» adicional, en este caso un «switch» de 5 puertos con un valor de 15€ y cables ethernet CAT 6 con un precio de 10€.

Otros recursos que hay que tener en cuenta son los servicios «básicos» como es la electricidad, cuyo coste sería un poco más de 16€ anuales tal y como se ha explicado en el apartado anterior de «Análisis energético» y el acceso a internet desde el hogar que normalmente viene incluido con un router. La velocidad mínima de internet sería de 100 megabits por segundo simétrico, aunque la recomendable para evitar cuellos de botella es de 1 gigabit por segundo. Dependiendo del proveedor de internet este precio varía mucho pero la media para 100Mbit/s es de unos 20€ mensuales y para 1Gbit/s 60€ mensuales.

En cuanto a recursos humanos se necesitarán aproximadamente 30 horas en investigación y 40 horas para configurar, probar y solucionar problemas en ambos sistemas.

CAPÍTULO 4

Diseño de la solución

4.1 Arquitectura del sistema

Como se ve en la figura 4.1 los componentes clave a nivel de red para este sistema son la Raspberry y el NAS. Ambos servidores pertenecen a la misma red local y por tanto los dispositivos que pertenezcan a esta pueden acceder a ellos, es por esto que, por ejemplo, un ordenador personal que pertenezca a esta red podrá conectarse al NAS sin necesidad de conectarse al servidor VPN.

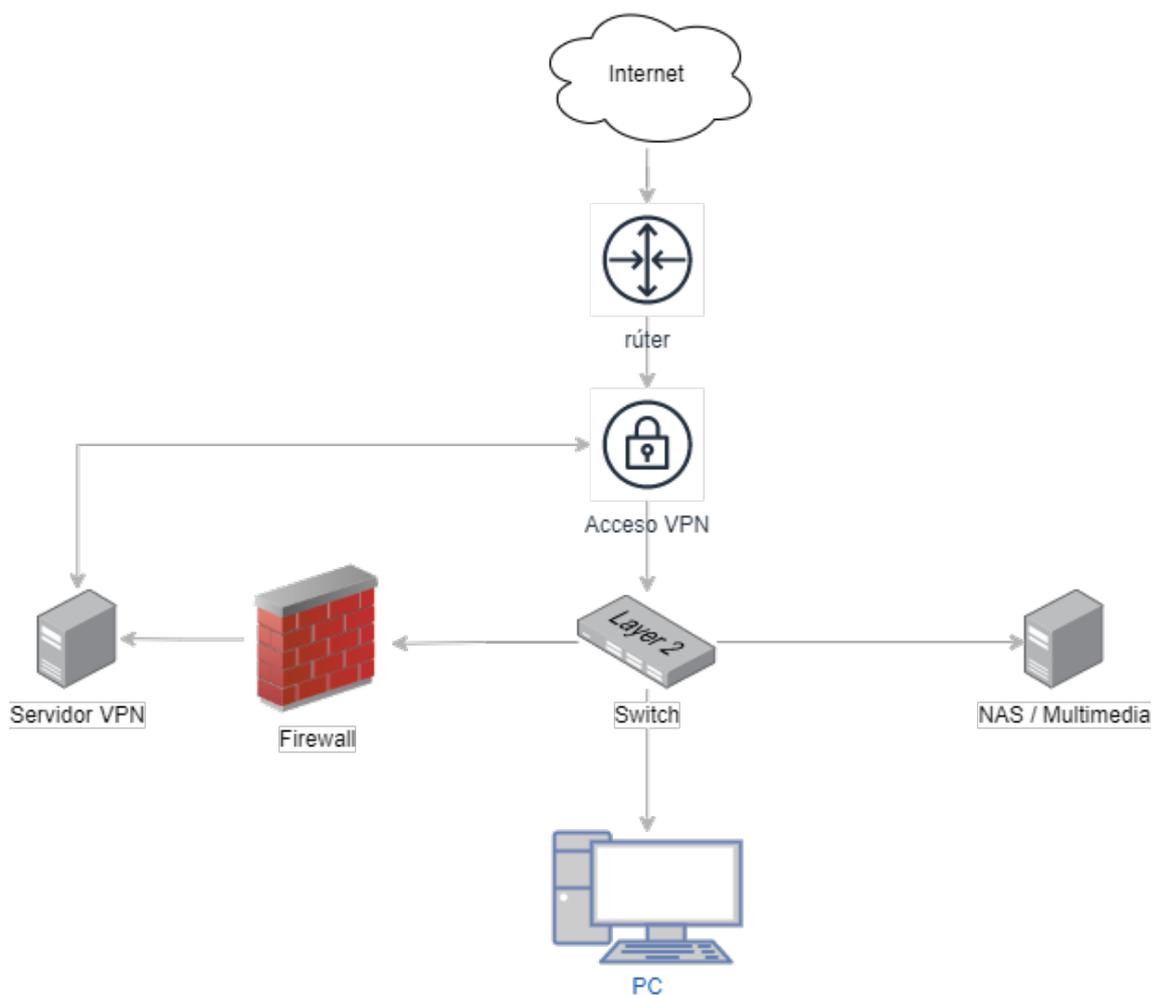


Figura 4.1: Arquitectura de la red

Por otro lado gracias a la VPN y al router, dispositivos que estén fuera de la red local se pueden conectar al NAS de forma segura, la propia VPN otorgará a estos dispositivos una dirección IP «virtual» de la red local y así estos podrán comunicarse con la red como si estuvieran en ella.

En cuanto al cortafuegos del servidor VPN de la figura 4.1 éste servirá para bloquear ciertas direcciones IP o patrones de ataque para así dar al sistema mayor seguridad y evitar el compromiso de la red. Este tema se abordará con mayor detalle en el capítulo de seguridad.

4.2 Diseño detallado

Tal y como se ha comentado en la sección anterior los componentes clave del sistema son tanto el servidor VPN como el servidor NAS. El servidor VPN será una Raspberry PI 4B con 4GB de memoria RAM DDR4 y con un procesador ARM de 4 núcleos, además dispone de USB 3.0 de 5Gbit/s y Ethernet de hasta 1Gbit/s.

Por otro lado el servidor NAS será un ordenador de sobremesa HP con 16GB de RAM DDR3 y con un procesador x86 APU (Accelerated processing unit) de 4 núcleos de CPU y 6 núcleos gráficos, también dispone de Ethernet de 1Gbit/s y puertos internos «SATA» de hasta 6Gbit/s para conectar los discos necesarios, para empezar en este proyecto se conectará un disco «HDD» de 1TB con velocidades de 120 MB/s, suficiente para saturar el puerto Ethernet de 1Gbit/s.

A parte de estos servidores existen más elementos de la red a mencionar.

Otro elemento que aparece en la figura 4.1 es el «switch». Los «switches» son dispositivos a nivel de enlace, nivel dos, que permiten conectar más dispositivos a una red y permite que estos puedan comunicarse entre si. En una red doméstica «normal» el modo de conectar los dispositivos es mayoritariamente inalámbrico gracias al Wi-Fi, pero muchos otros dispositivos necesitan conexión por cable «ethernet», ya sea por no disponer de tarjeta de red para Wi-Fi o por necesitar un mejor rendimiento que el Wi-Fi no ofrece, y no siempre el router dispone de suficientes puertos «ethernet» para todas las máquinas. Por tanto teniendo en cuenta que vamos a añadir otros dos dispositivos a la red «ethernet» es necesario añadir un «switch», a ser posible de la misma velocidad que la red, en este caso un «switch» de 1Gbit/s por cada puerto.

También mencionar la importancia de un buen cableado de la red, no sirve de nada esforzarse en tener dispositivos con tarjetas de 1Gbit/s de velocidad si después el cable no está capacitado para esas velocidades. Para no tener ningún problema ahora ni en un futuro próximo todo el cableado es de categoría 6 (CAT6), que soporta 1Gbit/s pero con el nuevo estándar 802.3z estos cables podrán soportar hasta velocidades de 5Gbit/s [8]. Por tanto de momento no existe una limitación a nivel de enlace.

Otro dispositivo es el router. Este es la «puerta al exterior», gracias a él la red se puede comunicar con internet y viceversa, pero supone un peligro muy grande ya que cualquiera podría acceder a los dispositivos de la red local. Es por eso que el router mantiene los puertos cerrados y los abre solo cuando son necesarios, pero esto supone otro reto, ¿Cómo se entra a la red sin abrir los puertos? La respuesta corta es que no se puede, se tiene que abrir los puertos para entrar en ella, pero existe una forma de abrirlos y al mismo tiempo no comprometer la red, «Port forwarding» (Reenvío de puertos). El «Port forwarding» ayuda a reenviar los paquetes que van destinados a un puerto del router a un dispositivo de la red local en concreto, en este caso ese dispositivo es el servidor VPN. De esta forma el servidor VPN solo espera por esos puertos peticiones VPN descartando el resto de manera

que los otros puertos del rúter permanecen cerrados. Posteriormente en el capítulo 6 se explicará con más detalle esta implementación.

CAPÍTULO 5

Tecnología utilizada

5.1 Sistema Operativo

La elección del sistema operativo (SO) es de suma importancia ya que este limitará en gran medida lo que podemos hacer. «Windows» y «Windows Server» son sistemas operativos que requieren licencias y por tanto no nos darían demasiada flexibilidad, es por razones que se usará Linux. Linux es solo una parte del SO lo que se conoce como el «kernel» o el núcleo, el resto del SO son paquetes software que se añaden al núcleo. Dependiendo de que paquetes se agreguen y que repositorios se usen tendremos una distribución u otra, también conocido como «flavour». La elección de esta distribución es lo que importa ya que, dependiendo de la que se escoja se tendrá acceso a un repositorio u otro y por tanto se tendrá un software u otro.

Las distribuciones más importantes se pueden clasificar en dos tipos según su gestor de paquetes:

- DEB : Debian ¹, Ubuntu ², Linux Mint ³...
- RPM : CentOS ⁴, Fedora ⁵, SUSE ⁶...

Para este trabajo las distribuciones que más flexibilidad y software tienen son las basadas «DEB» (basadas en Debian).

5.1.1. SO para servidor VPN

El servidor VPN se instalará en la Raspberry PI, este es un ordenador de características similares a un teléfono móvil, está diseñada para consumir muy poca energía con un rendimiento inferior a ordenadores de sobremesa. Esto es principalmente por la arquitectura ARM en la cual se basa este ordenador.

Por tanto para esta arquitectura especial también se necesita un sistema operativo especializado para este tipo de sistemas, «Raspberry PI OS» o también conocido como Raspbian. Como se puede intuir por este segundo nombre este SO también está basado en la distribución Debian, específicamente en Debian 11. Esta distribución, al igual que

¹<https://www.debian.org/>

²<https://ubuntu.com/>

³<https://linuxmint.com/>

⁴<https://www.centos.org/>

⁵<https://getfedora.org/es/>

⁶<https://www.suse.com/>

Debian 11, solo instala los programas necesarios para que el sistema funcione pero añade elementos que solo la Raspberry PI tiene para que puedan ser controlados a nivel de SO, como por ejemplo: los pines de GPIO («General Purpose Input/Output»), el control de leds o la lectura de temperaturas de la CPU. Además está diseñado para mejorar la eficiencia de sistemas con poca RAM y velocidades de CPU lentas, en este caso la CPU alcanza hasta 1,5GHz, algo lento para los estándares de hoy en día.

5.1.2. SO para el servidor multimedia

El servidor multimedia debe ser capaz de tener suficiente potencia para codificar el vídeo cuando se haga «streaming» es por esto que instalar un SO que utiliza demasiados recursos no tendría sentido.

Openmediavault es una distribución basada en Debian diseñada sobretodo para sistemas NAS, esta ya incorpora muchos paquetes software que se van a necesitar por ejemplo OpenSSH, SMB o incluso un panel de control web propio. La desventaja por otro lado es que también tiene instalado otros muchos programas que no se necesitan o que incluso pueden entorpecer a la hora de instalar y configurar los que se necesiten instalar.

La filosofía de Debian 11 es completamente distinta a Openmediavault ⁷, en vez de ofrecer una suite de programas ya preinstalados ofrece un SO casi vacío con programas básicos como una terminal Bash, escritorio gráfico (si así se le indica) y OpenSSH ⁸. Esto tiene un beneficio muy importante que es ayudar al rendimiento del sistema y la seguridad del mismo. Esto se consigue instalando solamente lo necesario, limitar un sistema a lo que necesita siempre es una buena práctica, la desventaja es que requiere más trabajo para instalar y configurar los paquetes software ya que hay que hacerlo manualmente.

Por lo tanto una vez visto las ventajas e inconvenientes de cada distribución se utilizará Debian 11 con sistema operativo para el servidor NAS.

5.2 SSH

SSH (Secure Shell) es un protocolo seguro a nivel de aplicación que funciona por TCP (Transmission Control Protocol). Sirve principalmente para la administración de servidores de forma remota a través de una terminal de órdenes pero también acepta interfaz gráfica gracias a que puede redirigir el tráfico del Sistema de Ventanas X o «X-Window System» en inglés.

SSH logra ser un protocolo seguro gracias al cifrado de la información entre cliente y servidor, esto se consigue por dos vías, uso de contraseña para iniciar sesión con un usuario en el servidor o gestionar claves RSA (Rivest-Shamir-Adleman) a través del intercambio de llaves públicas Diffie-Hellman [10] [9].

Este intercambio funciona utilizando una pareja de claves, una pública y otra privada, esta pareja la crea el servidor y este transmite su clave pública a los usuarios que se autenticen correctamente. Con esta llave pública el usuario/cliente puede cifrar información para que solo el servidor con su clave privada pueda descifrarla y así conseguir una comunicación segura. [11]

Cuando se instale el servidor «OpenSSH» ⁹ en el capítulo de «Desarrollo de la solución propuesta» se verá de forma práctica este intercambio de llaves.

⁷<https://www.openmediavault.org/>

⁸<https://www.openssh.com/>

⁹<https://www.openssh.com/>

OpenSSH, es la implementación de código abierto del SSH que se utiliza en Linux.

5.3 SMB

«Server Message Block» (SMB) es un protocolo que sirve principalmente para transmitir archivos a través de la red aunque también sirve para compartir impresoras y más elementos. Funciona a nivel de aplicación y utiliza TCP como protocolo de transporte. Este protocolo es similar a HTTP ya que funciona por peticiones (REQUEST) y respuestas (RESPONSE) donde el cliente hace las peticiones y el servidor las respuestas.[13]

La seguridad de SMB recae principalmente en la autenticación de usuarios con contraseña aunque si se indica en la configuración se puede cambiar esta autenticación y cifrar la conexión utilizando distintos algoritmos entre los que se recomiendan AES para cifrado y Kerberos para la autenticación. A parte de la autenticación y el cifrado también se deben indicar los permisos del usuario para que este solo pueda ver sus archivos, esto es muy importante para la confidencialidad.[12]

SMB es un servicio integrado en los sistemas operativos de Microsoft ¹⁰, al trabajar con un sistema operativo Linux, se utilizará una versión «open source» para sistemas UNIX llamada SAMBA ¹¹ que es compatible con SMB.

5.4 Comparativa VPNs

La VPN es un elemento clave para este proyecto porque dependiendo de cual se escoja se obtendrá mayor o menor rendimiento y mayor o menor seguridad.

Esta tecnología sirve, como su nombre indica, para crear redes privadas virtuales, es decir, sirve para poder “unir” dos redes locales (o dispositivo y red) que no tiene una conexión directa. Esta conexión se consigue utilizando protocolos que puedan crear “túneles” entre redes o canales seguros para que se puedan pasar información entre ellas sin riesgo de pérdidas de paquetes ni tampoco en comprometer ni su integridad ni confidencialidad.

Principalmente existen dos tipos, «Remote-Access VPN» y «Site-to-Site VPN».

«Remote-Access VPN» es básicamente cuando un «host» se conecta a una red distinta a la que se encuentra. Esto es muy útil para usar en dispositivos móviles que funcionan por redes «LTE» o dispositivos que se encuentran en redes a las que no se tiene acceso al rúter.

En cambio «Site-to-Site VPN» es una conexión entre dos rúteres de dos redes distintas. Este método sirve principalmente para grandes empresas que quieran conectar sucursales entre sí.

En el caso de este proyecto utilizaremos «Remote-Access» por la necesidad de poder conectarse desde cualquier lugar al servidor de nuestra red.

Desde el punto de vista de la seguridad, por un lado la información a transmitir (audios, vídeos, imágenes, etc) no es extremadamente sensible pero es información personal. Por esta razón la confidencialidad debe ser robusta para poder mantener la privacidad de los usuarios. Por otro lado debemos asegurar la autenticidad de los usuarios que se conectan a la VPN ya que al conectarse a esta se tiene acceso a la red y usuarios con malas intenciones podrían comprometer la red.

¹⁰<https://www.microsoft.com>

¹¹<https://www.samba.org/>

Además de la seguridad, el rendimiento también debe ser muy bueno ya que la resolución mínima del «streaming» en diferido, el servicio que más recursos consumirá, será «High Definition» (HD) o 1280 x 720 píxeles por fotograma pudiendo aumentar hasta «4K ultra high definition» (4K UHD) o 3840 x 2160 píxeles por fotograma si se trata de películas. Es por esto que el ancho de banda de la VPN debe ser suficiente para que la reproducción del vídeo se produzca sin cortes. Este ancho de banda de la VPN dependerá de la cantidad de información útil que se envíe por paquete, la cantidad de procesamiento necesario para el cifrado y la fiabilidad de la red/protocolo usado, suponiendo que no hay más cuellos de botella que la propia VPN.

Esta información útil por paquete dependerá del tamaño de las cabeceras extra que se añaden a las tramas «Ethernet» para poder utilizar la VPN, es decir cuanto más espacio ocupen las cabeceras en una trama «Ethernet» (máximo por trama 1518 B) menos espacio se tendrá para la información útil, por tanto la velocidad de transferencia también se reduce.

Teniendo en cuenta la información anterior, se va a realizar una comparativa entre las tecnologías de VPN más populares para poder obtener la que mejor se adapte a las necesidades del proyecto.

5.4.1. PPTP

PPTP (Point to Point Tunneling Protocol) se ha descartado casi de inmediato debido a sus problemas de seguridad ya que se ha conseguido romper el protocolo en varias ocasiones y por tanto se conocen sus puntos débiles, incluso existen guías de como hacerlo [14], además de que se trata de una tecnología antigua y obsoleta.

Sí que es verdad que al tratarse de un protocolo que utiliza cifrados muy débiles no necesita mucho poder de computo y como tampoco utiliza muchas cabeceras (40 B extra) hay más información “útil” por trama y el ancho de banda no se reduce tanto, es por esto que PPTP es una VPN muy rápida.

Igualmente la falta de seguridad hace que esta tecnología no sea recomendable.

5.4.2. Ipsec/L2TP

L2TP (Layer 2 Tunneling Protocol) es la "versión"segura de PPTP, esta versión resuelve los problemas de seguridad de PPTP apoyándose en IPsec[18] como protocolo para cifrar y autenticar la integridad.

Al usar IPsec la seguridad de la VPN está garantizada porque se proporciona integridad, confidencialidad y autenticidad. Para la integridad y autenticidad del paquete se suele usar HMAC-SHA1/2 [20]. Para la confidencialidad, es decir el cifrado, el algoritmo más usado es AES (Advance Encryption Standard) con distintas modificaciones como AES-CBC o AES-CCM[19] y por último para autenticación normalmente se usa los algoritmos de RSA (Rivest-Shamir-Adleman) o PSK (Pre-Shared Key), aunque en este caso como la negociación se tendría que hacer con el protocolo CHAP (Challenge Handshake Authentication Protocol) de L2TP, un protocolo poco seguro, se decidió utilizar otro protocolo para poder establecer una «Security Association» o SA. Este protocolo es IKE o «Internet Key Association», el cual se explicará en el apartado 5.4.4 Ipsec/IKEv2.

En cuanto al rendimiento, este dependerá del protocolo de cifrado elegido siendo AES el más rápido de ellos. Y además al tratarse de un túnel a nivel de enlace («Layer 2») puede aumentar su rendimiento aunque también puede dar problemas ya que al tratarse de un

protocolo a nivel de enlace también puede dar problemas con las NAT (Network Address Translation).

Por tanto esta VPN cumple con las necesidades de seguridad y rendimiento teniendo como pega que L2TP se trata de un protocolo algo antiguo y aún usa CHAP. Además los proveedores de internet a veces bloquean cabeceras ESP (Encapsulating Security Payload) de IPsec de manera que cuando esto pasa la VPN no es funcional.

5.4.3. OpenVPN

OpenVPN ¹² utiliza la biblioteca de software OpenSSL ¹³ que implementa una versión libre de SSL/TLS (Secure Socket Layer)/(Transport Layer Security), tecnología que ya se usa para HTTPS, consiguiendo así un estándar de seguridad y cifrado muy elevado. Además para la autenticación se pueden utilizar claves precompartidas o certificados lo que lo convierte en una tecnología muy segura.

En cuanto al rendimiento se trata de una VPN rápida, sobretodo en su modo UDP Extensible Authentication Protocol(User Datagram Proocl), el cual favorece la velocidad a costa de la fiabilidad. Esta velocidad depende en gran medida del cifrado que se use, siendo AES un cifrado rápido y seguro, tal y como se ve en la figura 5.1, donde se puede ver que la VPN con cifrado tiene cifras similares a la VPN sin cifrado usando OpenVPN.

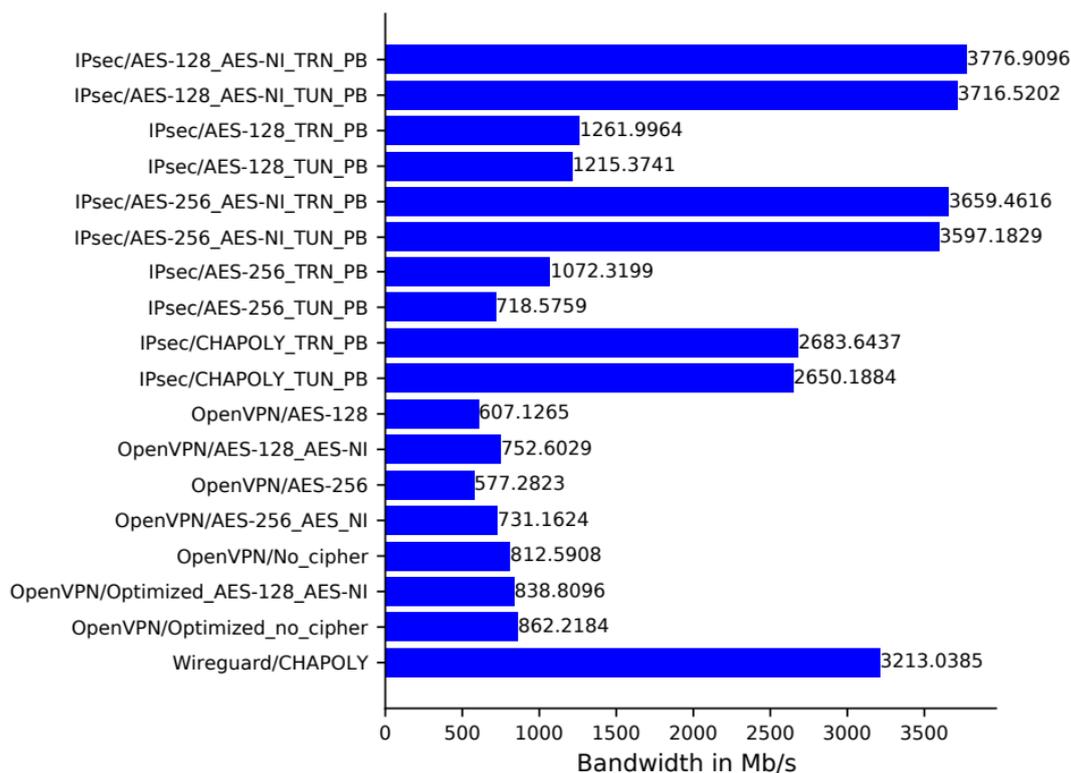


Figura 5.1: Gráfico rendimiento OpenVPN con distintos cifrados[16]

La mayor desventaja de esta tecnología es que se encuentra integrada en muy pocos dispositivos y es necesario aplicaciones de terceros para su uso. Por otro lado la dificultad para instalar y configurar esta VPN es bastante elevada, además de que el «handshake» o

¹²<https://openvpn.net/>

¹³<https://www.openssl.org/>

el tiempo que se tarda en crear la conexión es bastante alto comparándolo con otras VPN como L2TP o IKEv2.

Por otro lado no deberemos preocuparnos por bloqueos por parte del proveedor de internet o problemas con la «NAT» porque esta VPN trabaja a nivel de aplicación, aunque esta ventaja tiene como precio la reducción del rendimiento de la VPN.

5.4.4. Ipsec/IKEv2

IKE (Internet Key Exchange) es un protocolo que se creó para poder establecer las SA (Security Association) de IPsec utilizando certificados del estándar «X.509» o comúnmente conocidos como formato de llaves públicas. IKE se base en otros dos protocolos ISAKMP y Oakley este último utiliza el famoso algoritmo de «Diffie-Hellman»[22].

Este protocolo funciona, de forma muy resumida, creando un secreto de sesión del cual se obtienen las llaves criptográficas a través del protocolo «Diffie-Hellman» y los certificados ya compartidos, es decir, para autenticar al usuario y crear las llaves simétricas de la sesión se utiliza un algoritmo pesado computacionalmente, y para transmitir la información se utiliza algoritmos criptográficos de llaves simétricas que son mucho más rápidos y eficientes[23].

Tal y como se ha comentado en el apartado sobre L2TP, IKE es un protocolo que se usa para la negociación de las «SA» de IPsec, IKEv2 es la versión mejorada de IKE pero que sigue sus mismos principios. Por tanto la seguridad de esta VPN es muy robusta ya que la confidencialidad, integridad y autenticidad de los paquetes son aseguradas por IPsec.

IPsec a su vez se compone de otros tres protocolos AH «Authentication Header», ESP «Encapsulating Security Payload» e IKE, ya explicado.

El protocolo AH es el que se encarga de garantizar la integridad y autenticidad del paquete, esto lo logra calculando un código hash de las partes importantes (contenido IP y clave secreta) y partes constantes del datagrama. Este hash puede resultar en un problema ya que no permite usar NAT.

ESP por otro lado se encarga principalmente de la confidencialidad, aunque también otorga integridad y autenticidad de los datos que se quieren transmitir y solo de estos dejando la cabecera IP desprotegida, además algunos proveedores de internet, como ya se ha comentado, pueden bloquearlo.

Para poder entender mejor como funcionan los protocolos de AH y ESP ver la figura 5.2.

También mencionar que IPsec tiene un modo túnel y un modo transporte, para utilizar VPN se tiene que usar este último modo.

Las mejoras que implementa IKEv2 sobre IKE son la adición de un estándar para mejorar la movilidad de la VPN, es decir mantener la conexión aunque cambie la red, resistencia a ataques de denegación de servicio DDoS (Denial of Service) y por último a destacar la agrupación de todos los RFCs, documentación del protocolo, que definían a IKE (3 RFCs distintos) en uno solo[22].

En cuanto a la eficiencia, esta VPN tiene un tiempo de «handshake» muy bajo, el ancho de banda se ve muy poco afectado por el tamaño extra de cabeceras y dependiendo del cifrado utilizado se perderá más o menos rendimiento. Es decir esta VPN está a la par con L2TP y OpenVPN en modo UDP (User Datagram Protocol) pero es más eficiente y tiene menor tiempo de establecimiento de conexión.

Además la configuración de esta tecnología es bastante sencilla aunque no tanto como PPTP.

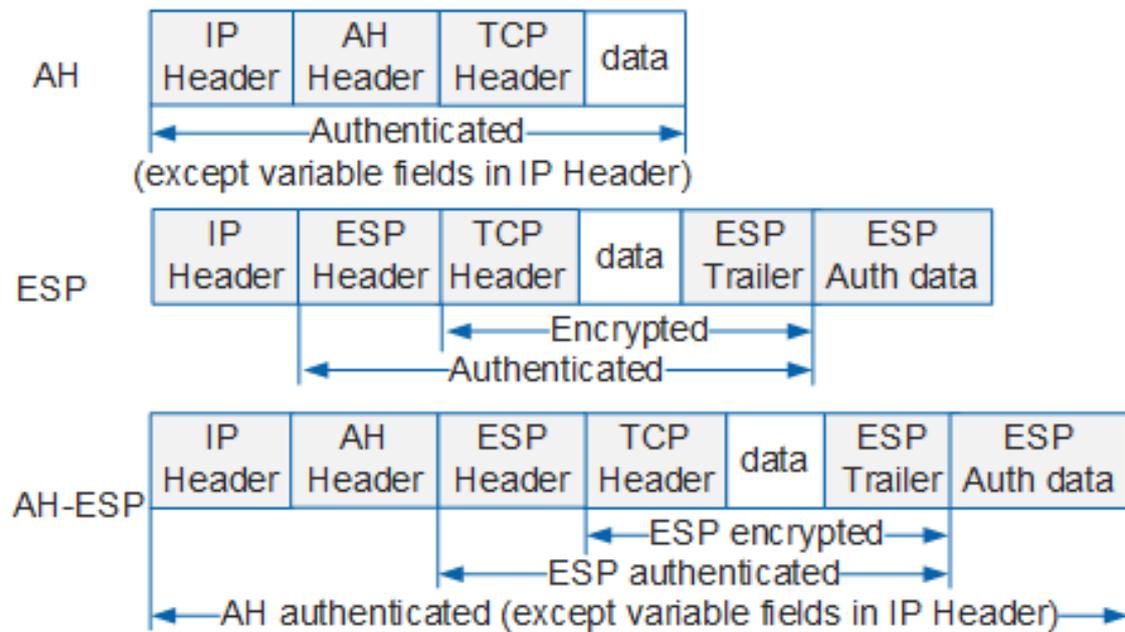


Figura 5.2: Datagramas IPsec en donde se aplican los distintos protocolos [21]

Su mayor desventaja es por la cabecera ESP de IPsec porque algunos proveedores de internet pueden bloquearlo. También pueden haber problemas con la NAT por culpa del AH, aunque esto se soluciona la mayoría de ocasiones porque IKEv2 tiene un sistema de detección de NAT y si la detecta hace uso de paquetes «keep-alive» para que el rúter del dispositivo no cierre la «apertura» en la NAT.

Este tipo de VPN ha sido probada de muchas formas por la NSA (Nation Security Agency) y según ellos consiguieron romper este protocolo asegurando que se podría romper el 66 % de todas la VPNs basadas en IKE, posteriormente este hecho fue refutado en varias ocasiones comprobando la seguridad de esta VPN[24][25] si se configura correctamente. Se hablará de este problema en el capítulo de «Seguridad». Igualmente IKE es susceptible a ataques de tipo diccionario si se utiliza una contraseña corta cuando se haga autenticaciones con EAP (Extensible Authentication Protocol).

Por estas razones esta VPN será la que se instalará en el sistema.

La versión «opensource» para Linux seleccionada es strongSwan¹⁴ un programa muy bien documentado que tiene ejemplos de uso y configuraciones, además de que dispone de soporte y foros para solucionar la mayoría de problemas.

5.5 Reproductor multimedia

Una vez se tiene el servidor y una conexión rápida y segura se debe de encontrar una forma de poder reproducir el contenido del servidor sin tener que descargar el vídeo entero porque sería un gasto innecesario de recursos y podría suponer problemas de compatibilidad ya que si un dispositivo no tiene el espacio necesario no podría ver el vídeo. Tal y como se ha comentado en el capítulo del contexto si se quiere ver contenido multimedia propio no se puede utilizar otros reproductores como Netflix¹⁵ o HBO¹⁶ ya que estos solo reproducen contenido que tiene almacenado en sus servidores.

¹⁴<https://www.strongswan.org/>

¹⁵<https://www.netflix.com/>

¹⁶<https://www.hbo.com/>

Además como se quiere tener la máxima compatibilidad posible se ha descartado SMB como reproductor, aunque permite el «streaming» de contenido sin la necesidad de descargarlo no todos los dispositivos son compatibles con este protocolo por ejemplo (y uno de los más importantes) las «Smart TVs» o televisores inteligentes.

Es por esto que se va a instalar un reproductor a través de internet propio, que tendrá acceso a los elementos del servidor y los podrá reproducir por protocolos de «streaming» tanto desde una aplicación propia o mediante una web.

Hay distintos reproductores famosos como Kodi ¹⁷, Plex ¹⁸ o Emby ¹⁹. Es este último el que se acabará utilizando en el proyecto.

5.5.1. Emby

Emby no es un software libre por lo que la comunidad no tiene acceso al código de la aplicación, pero sigue siendo gratuito solo que existe una versión con mayores funcionalidades que es de pago. Este software permite reproducir casi todos los formatos multimedia: vídeos, imágenes, audio, «streaming» en vivo ...

A parte de ser un reproductor también dispone de un servidor web propio lo que proporciona una compatibilidad casi perfecta ya que cualquier dispositivo con un navegador es capaz de utilizar este software. Además dispone de aplicación propia que se puede instalar en casi todas las plataformas, incluyendo la mayoría de marcas de «Smart TV», haciendo la experiencia menos frustrante, y si lo anterior no fuera suficiente este software también dispone de un servidor propio de DLNA (Digital Living Network Alliance). El DLNA sirve para usar las mismas funcionalidades que con un dispositivo chromecast pero sin tenerlo, es decir, permite el streaming en directo desde el servidor Emby hasta el dispositivo donde se quiere reproducir.

Por otro lado este software tiene la capacidad de poder tener varios usuarios y poder seleccionar que puede ver cada uno de ellos pudiendo así crear perfiles infantiles o perfiles especializados en un contenido multimedia en específico.

Otra característica interesante de Emby es que permite codificar la resolución del vídeo que se está viendo en ese momento permitiendo así ajustar la calidad al ancho de banda que tenga el usuario.

En cuanto al rendimiento de Emby en su versión gratuita carece de aceleración por hardware, es decir, que no se puede utilizar una gráfica para aumentar el rendimiento del servidor para codificar vídeos. Este acepta muchos tipos de codificación y en la mayoría de casos no se necesitara codificar nada.

Por otro lado Emby no “sufre” tanto, comparándolo con Plex, cuando se reproducen vídeos de alta resolución en un formato MKV (Matroska), esto es porque Plex realmente no acepta este tipo de formato y lo codifica a MP4 al momento, esto provoca cortes en la reproducción y/o que no se permita avanzar o retroceder en el vídeo, por tanto lo mejor para evitar estos problemas es codificar previamente los vídeos a MP4[26], cosa que es una gran molestia. En cambio Emby solo codifica el vídeo si el cliente no acepta la codificación por defecto. Esta es una de las principales razones por la que se ha escogido Emby frente a Plex.

Por tanto se trata de un reproductor con muchas cualidades y con una compatibilidad casi perfecta y que permite adaptarse a la circunstancias de la red.

¹⁷<https://kodi.tv/>

¹⁸<https://www.plex.tv/>

¹⁹<https://emby.media/>

5.6 Panel de control

Los paneles de control no son necesarios para administrar un sistema pero sí que son muy útiles ya que estos ofrecen una interfaz gráfica para poder controlar y configurar las características más importantes del servidor.

Además el panel de control también avisa del estado del sistema ya sea mostrándolo gráficamente o, si se configura, enviando correos electrónicos al administrador del sistema.

Por otro lado muchos paneles de control son servidores web, es decir, que la propia interfaz del panel está alojada en una página web que es accesible si se autentifica como un usuario con suficientes permisos. Es por esta última razón por la que se plantea la instalación de un panel de control porque al alojarse en una web es accesible desde cualquier navegador y de esta manera podemos controlar los servidores sin necesidad de conectarse por SSH (Secure Shell), algo que requiere una aplicación aparte en la mayoría de dispositivos móviles.

Entre la gran variedad de paneles de control web se ha seleccionado Webmin²⁰ porque se trata de un software de código libre con buena documentación y con la capacidad de añadir muchas características dependiendo de que aplicaciones software tenga el sistema, por ejemplo, si se añade el software SAMBA, Webmin automáticamente añadirá una nueva interfaz para poder configurarlo sin la necesidad de tener un terminal.

Igualmente destacar que la instalación de un panel de control es totalmente opcional y que en algunos casos es incluso contraproducente porque una interfaz gráfica puede retrasar a usuarios avanzados que estén muy familiarizados con el terminal.

5.7 DDNS

DNS o «Domain Name Server» es un tipo de servidor que ayuda a la traducción de una dirección IP en nombres más familiares. Esta traducción es estática, es decir, si la IP se asocia a un nombre cuando esta cambie se perderá esta asociación, es por eso que la mayoría de dominios tienen una dirección IP estática, una dirección que no cambia.

El problema es que la mayoría de direcciones IP domésticas que ofrecen los proveedores de internet son dinámicas, cambiar a mano la asociación cada vez que la IP cambia es algo inviable, es por esto que un servidor DNS convencional no serviría, se necesita uno dinámico.

DDNS o «Dynamic Domain Name Server» es un servidor DNS pero que tiene la capacidad de crear y actualizar asociaciones nombre-IP. Estas actualizaciones empiezan por parte del cliente el cual envía periódicamente (o cuando se detecta un cambio) una petición al servidor DDNS con el nombre de dominio de la asociación, su nueva dirección IP y una clave que sirve para autenticarse. Una vez el servidor tiene los tres datos actualiza la asociación.

De esta forma no importa que la dirección IP cambie ya que el dominio será estático.

En este caso al ya tener un dominio comprado en IONOS²¹ se va a utilizar su propio servidor DDNS a través de su API.[29]

Igualmente existen multitud de opciones completamente gratuitas que ofrecen el mismo servicio, por ejemplo duckdns.org, o incluso la opción de alojar un DDNS en una nube con IP estática propia de forma gratuita gracias a Oracle Cloud y sus máquinas

²⁰<https://www.webmin.com/>

²¹<https://www.ionos.es/>

virtuales «Always Free»²², como la de la figura 5.3 que está formada por 1 núcleo de CPU y 1 GB de RAM, recursos más que suficientes para un DDNS.

Para esta última opción el software ideal sería Bind9, un DNS que permite la actualización automática de registros convirtiéndose así en DDNS [28]. Esta opción no se desarrollará más porque no es el objetivo de este trabajo pero era importante mencionarla porque es una solución gratis y muy personalizable.



Name	State	Public IP	Private IP	Shape	OCPU count	Memory (GB)
host1 <small>Always Free</small>	● Running	[REDACTED]	10.0.0.230	VM.Standard.E2.1.Micro	1	1

Figura 5.3: Máquina Virtual de «Oracle Cloud»

5.8 Wake on LAN

WOL (Wake On Lan)[27] es un estándar de Ethernet y sirve para poder encender ordenadores a través de mensajes por la red.

Este estándar envía un paquete especial, denominado «Magic packet», a todos los dispositivos de la subred (broadcast), este contiene la dirección MAC del dispositivo que se quiere encender, los dispositivos que tengan activado WOL escucharán este tipo de paquetes, y si la dirección MAC coincide con la del propio dispositivo, este procederá a encenderse.

Todos los ordenadores que tengan activado el «Wake On Lan» dejarán en modo de escucha su tarjeta de red en el puerto 9 cada vez que se apaguen, este modo consume muy poca energía y, obviamente, solo funciona si el ordenador está conectado a la corriente.

De esta forma es posible encender de forma remota los dispositivos que se quiera cuando se necesite.

²²<https://www.oracle.com/es/cloud/free/>

Desarrollo de la solución propuesta

Como ya se ha comentado anteriormente este sistema consta de dos ordenadores, una Raspberry PI y el NAS. La Raspberry PI será la encargada de mantener la mayoría de procesos ya que estos no consumen muchos recursos, estos procesos son: servidor VPN, servidor WEB, panel de control del propio ordenador y diversos «scripts» que usan WOL.

Por otro lado el ordenador de sobremesa ejecuta un menor número de procesos, pero estos consumen una cantidad mucho mayor de recursos, esto procesos son: servidor multimedia, servidor SMB y su propio panel de control.

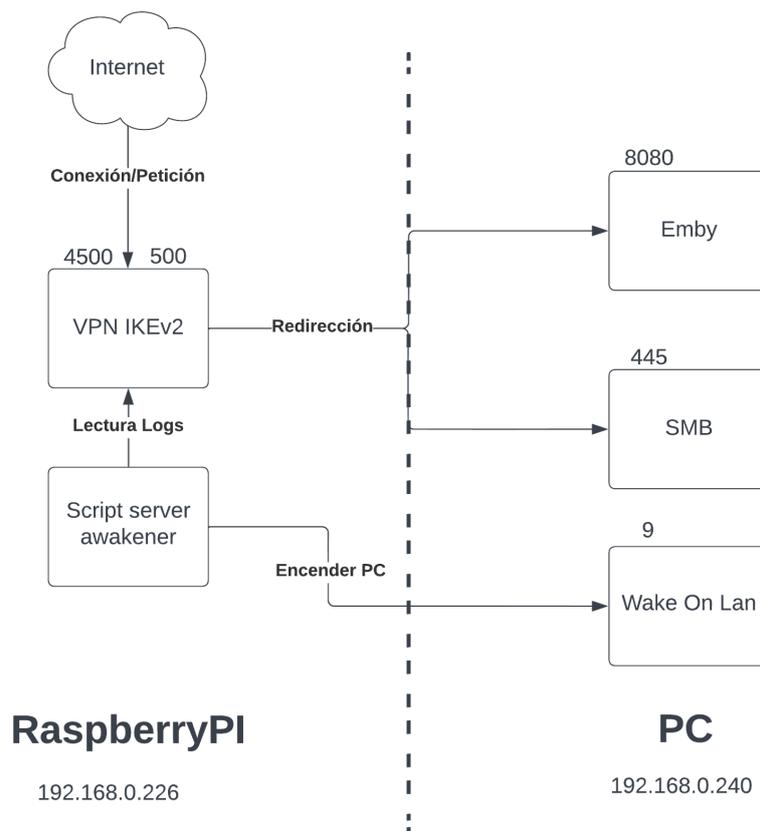


Figura 6.1: Diagrama de relación entre procesos del sistema

La relación entre servidores es sencilla, la Raspberry PI se encargará de encender el ordenador de sobremesa a través de los «scripts» que usan WOL cuando detecte que algún usuario se ha conectado al servidor VPN o se haya despertado desde la web, que aloja

ella misma. Una vez despertado el PC la propia Raspberry PI redirigirá las peticiones que vienen desde la VPN a los procesos correspondientes.

En el diagrama de la Figura 6.1 se puede ver de mejor manera la relación entre los procesos y que puertos se abren en cada uno de ellos para poder comunicarse.

6.1 Montaje de los ordenadores

6.1.1. Instalación del ventilador para la Raspberry PI

La Raspberry PI es un ordenador muy eficiente que para procesos ligeros no necesita una refrigeración activa como un ventilador sino que con disipadores de calor de aluminio se consigue refrigerar las partes más calientes. Igualmente, aunque no sea obligatorio, se recomienda tener un ventilador para tener mejor controladas las temperaturas del dispositivo.

El ventilador se conectará a los pines de la Raspberry PI, por tanto el voltaje del ventilador deberá ser igual al máximo voltaje que estos pines pueden transmitir, en este caso son 5 voltios como se puede observar en la figura 6.2.[30]

En este caso utilizaremos un ventilador de 40 milímetros de lado de 5 voltios y un adaptador de 3 pines a 2 pines tal y como se ve en la figura 6.3.

Estos conectores, rojo (positivo) y negro (negativo), se colocarán en los pines del GPIO número 4 (+5V) y 6 (GND) respectivamente que aparecen en la figura 6.2.

Con este añadido tal y como se ha dicho antes conseguiremos prevenir posible sobrecalentamientos e incluso aumentaremos la vida útil del silicio.

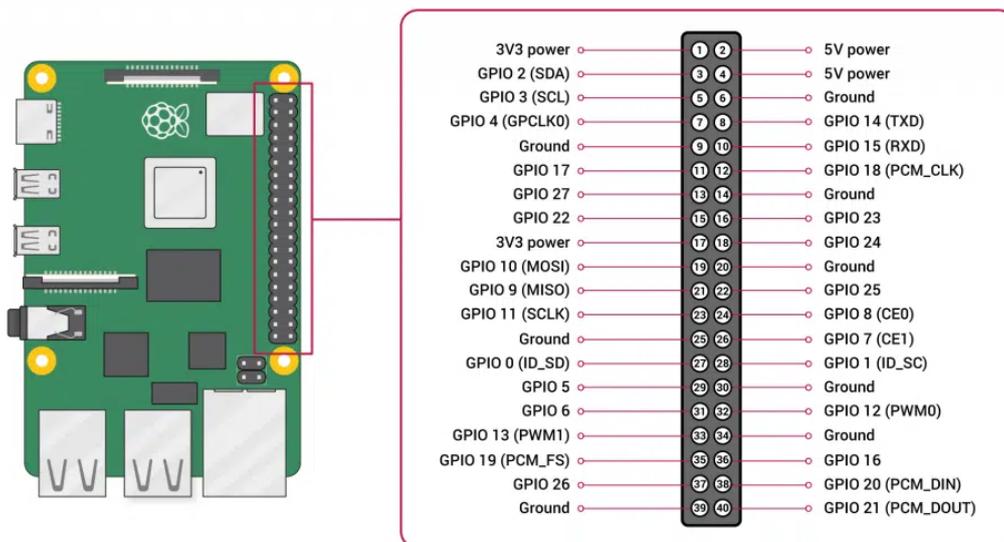


Figura 6.2: Tabla de pines y voltajes del GPIO[31]



Figura 6.3: Ventilador y cable de adaptación

6.2 Instalación sistema operativo

La instalación de los sistemas operativos de ambos dispositivos es muy sencilla y por tanto solo se explicará la instalación de Raspberry PI OS ya que se hará de forma remota.

6.2.1. Raspberry PI OS modo «headless»

Esta instalación es muy sencilla ya que es un sistema operativo diseñado para instalarse automáticamente, para este proyecto se instalará en modo «headless», es decir, sin tener que conectar periféricos directamente al dispositivo. Para ello seguiremos la guía oficial [32].

Para la instalación se necesitará una tarjeta SD y un lector, en este caso se ha escogido una de 32GB, esta tarjeta será la unidad de almacenamiento del dispositivo y es donde se instalará el SO con la ayuda de un software. El software que se utilizará es Etcher, este formateará y preparará la imagen del SO en la tarjeta. Una vez completado este proceso se deberá añadir un archivo sin extensión que se llamará SSH en la carpeta raíz de la SD tal y como se ve en la figura 6.4, de esta forma se habilitará el encendido automático del servidor «OpenSSH» y se podrá acceder a la Raspberry PI mediante un cable Ethernet. Para poder conectarnos con SSH necesitaremos un usuario con contraseña y la dirección

kernel8.img	31/03/2022 19:40	Archivo de image...	7.971 KB
LICENCE.broadcom	31/03/2022 19:40	Archivo BROADC...	2 KB
ssh	12/05/2022 18:53	Archivo	0 KB
start.elf	31/03/2022 19:40	Archivo ELF	2.897 KB
start_cd.elf	31/03/2022 19:40	Archivo ELF	783 KB

Figura 6.4: Archivo SSH creado en el directorio raíz de la tarjeta SD.

IP del dispositivo. Obtener el usuario es fácil, existe uno por defecto llamado “pi” con contraseña “raspberrypi”. En cambio obtener la dirección IP es más complicado, existen distintas formas de hacerlo, en este caso el router detecta el nuevo dispositivo y le otorga una IP por «DHCP» mostrándola en la página web del mismo, figura 6.5. Una vez se establezca una conexión al dispositivo se deberá configurar una dirección IP estática, como ejemplo se ha usado 192.168.0.226, para saber más revisar el apéndice A.1. De esta forma la Raspberry PI siempre tendrá la misma dirección y será más fácil para conectarse a ella.

Para acabar la configuración inicial deberemos crear un usuario de uso normal y añadirlo al grupo «root» para permitir usar funciones de este grupo, además se eliminará

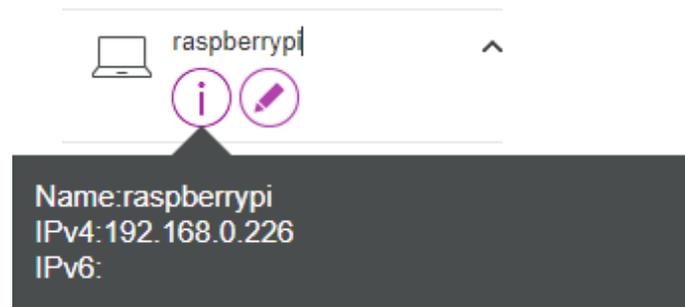


Figura 6.5: Interfaz del router donde se muestra el «hostname» y la IPv4 otorgada.

el usuario «pi» y se cambiará la contraseña del usuario «root» por temas de seguridad, visitar el apéndice A.2 para más información.

Con esto el sistema queda listo para poder trabajar en él.

6.3 Instalación software

6.3.1. VPN

En este apartado se explicará de forma muy resumida y sin profundizar como instalar la VPN, si se quiere más información revisar el apéndice A.3

Primeramente se instalarán los paquetes de strongSwan y librerías extra para poder usar EAP como método de autenticación para los usuarios.

Se crearán una pareja privada/pública (llave/candado) de certificados, la parte privada (server-cert.pem), la llave, se la quedará el servidor y la pública (ca-cert.pem), el candado, se transmitirá a los usuarios, de esta forma los usuarios se podrán autenticar en el servidor VPN de forma segura. Estos certificados requieren la creación de llaves privadas previas para “firmarlos”. Para entender mejor la relación entre los certificados y las llaves ver la figura 6.6. Hay que tener en cuenta que estos certificados se ligan a la IP o el dominio que se indica, por tanto si estos cambiaran los certificados no funcionarían y no se podría conectar a la VPN.

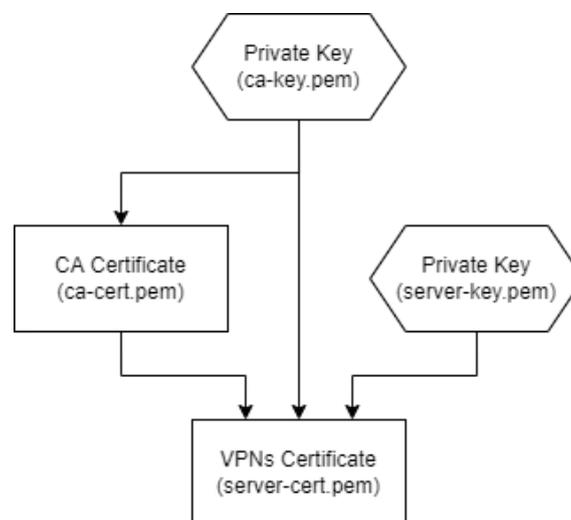


Figura 6.6: Diagrama de las dependencias entre certificados y llaves.

Una vez creados todos los certificados necesarios se procederá a configurar strongSwan en el archivo `/etc/ipsec.conf`, en este archivo se configurará tanto la VPN como el funcionamiento general del software. En esta configuración se especificará, entre otras cosas, que protocolo para obtener el «SA» usaremos, que algoritmos de cifrado e integridad se usarán, que direcciones IP aceptaremos y que IP virtual le daremos a los usuarios que se conecten entre otras cosas.

Y por último se deberá crear los «secretos» de EAP, PKI u otro protocolo dependiendo de que se haya elegido, estos «secretos» se pondrán en `/etc/ipsec.secrets` y se deberá especificar que protocolo se usará, nombre del usuario y contraseña. Hay que mencionar que los usuarios son de una sola conexión concurrente, es decir, si otro cliente usa el mismo nombre de usuario mientras este ya esté en uso, el nuevo cliente desconectará al antiguo echándolo de la VPN. Para evitar este problema lo mejor es utilizar un nombre de usuario distinto por cliente.

Una vez terminada de configurar la VPN en el servidor se tendrá que configurar el cliente, para ello se instalará el certificado de autoridad (certificado público), creado anteriormente en el servidor, en el dispositivo del cliente. Una vez instalado el certificado se configurará la VPN.

En dispositivos Windows7 o posteriores no hace falta instalar ningún software extra porque Windows ¹ ya tiene integrado IKEv2, aunque posteriormente se verá ciertos problemas con este sistema operativo 7.1.1, en clientes Linux dependerá del sistema operativo pero todos las distribuciones soportan los paquetes oficiales para esta VPN y simplemente es necesario instalar librerías extra para soportar EAP [35], por otro lado en móviles Android ² esta VPN ya viene integrada con la mayoría de protocolos de autenticación de usuario excepto con el protocolo EAP que es el que se va a utilizar, es por esto que se necesita instalar una aplicación de strongSwan, en cambio en dispositivos Apple ³ IKEv2 viene instalada en todos ellos sin necesidad de instalar aplicaciones ni librerías extra para poder usar EAP.

Si no se pudiera utilizar ninguna interfaz gráfica, existe un programa desarrollado por strongSwan llamado “charon-cmd” que es un cliente para conectarse a VPNs de tipo IPsec con IKEv1/2 por línea de comando. De momento solo está disponible para Linux [43].

Elección de algoritmos

Como se ha comentado en el apartado 5.4, para este proyecto es necesario una VPN eficiente y segura, es por eso que la elección de los algoritmos es muy importante ya que estos son los que aseguran la conexión y determinan en gran medida la velocidad de la VPN.

Existen dos tipos de algoritmos que se usan en esta VPN para la transmisión de paquetes de datos, de integridad y de cifrado.

Para el cifrado se ha usado una variación de AES, AES 256 bits con Galois Counter Mode 16, ya que se trata de un algoritmo muy rápido con bajo coste computacional e incluso más seguro que AES256 [47] [36]. Este modo solo estará disponible para dispositivos distintos de Windows, esto es debido a un problema que existe en este sistema operativo que se verá en el apartado 7.1.2

En cambio para la integridad el algoritmo que se usa es una versión eficiente y sin colisiones de SHA, el SHA384. Con una cantidad tan grande de bits se asegura que no se

¹<https://www.microsoft.com/es-es/software-download>

²<https://www.android.com/>

³<https://www.apple.com/>

producirán colisiones como ocurre con SHA-1 o MD5, ya que estas colisiones significarían que dos paquetes distintos podrían tener la misma "firma" y por tanto la VPN sería susceptible a ataques de «Men In The Middle»(MITM) donde el atacante recibiría los paquetes, los modificaría y los reenviaría. Al haber colisiones ambos mensajes tendrían la misma firma y por tanto no se podría saber si el mensaje recibido es el original.[36]

Los algoritmos expuestos son los deseados, es decir, los que el servidor VPN recomendará por encima de todo. En cambio si los clientes no aceptan estos algoritmos se pueden usar otros que hayamos puesto para aumentar la compatibilidad. Estos algoritmos se enseñan en el apéndice A.3.

6.3.2. SMB

SAMBA será el software que proporcione el protocolo SMB que permitirá la transferencia de datos entre servidores. Este software es muy fácil de instalar y configurar, revisar apéndice A.4. Las principales acciones que se deben realizar para que SAMBA funcione correctamente son configurar las particiones para los usuarios y añadir a la base de datos de SAMBA estos usuarios con su respectiva contraseña para que puedan iniciar sesión. Es importante mencionar que aparte de añadir los usuarios a la base de datos estos deben existir en el sistema con ese mismo nombre.

A parte de la partición de los usuarios también se creará una partición "pública" para el servidor multimedia, de esta forma cualquier usuario con acceso a la VPN puede subir su contenido multimedia y utilizar el servidor multimedia.

La idea principal es que cada usuario tenga su propia partición privada y que todos los usuarios puedan acceder y usar la partición pública.

6.3.3. SSH

SSH será la herramienta principal de administración de los servidores. Se instalará con el gestor de paquetes APT.

Para que este sea seguro es necesario utilizar una de dos opciones, uso de contraseña y usuario o intercambio de llaves, siendo más seguro este último.

En este caso se usará el intercambio de llaves no solo por seguridad sino también por comodidad ya que de esta forma no será necesario poner contraseña cada vez que se quiera acceder.

El cifrado de los mensajes de SSH con claves públicas (intercambio de llaves) Diffie-Hellman, se puede realizar tanto con RSA como con «Elliptic Curve». Una vez las claves están compartidas el cliente ya se puede autenticar de la forma en la que se ve en la figura 6.7.

Este envío de claves se puede hacer a mano, copiando y pegando las llaves públicas en los respectivos dispositivos, o mediante una herramienta incluida en «OpenSSH» llamada «ssh-copy-id» la cual solo necesita autenticarse una vez con contraseña para enviar la clave pública del cliente al servidor.

Para obtener información más detallada consultar el apéndice A.5

SSH Authentication

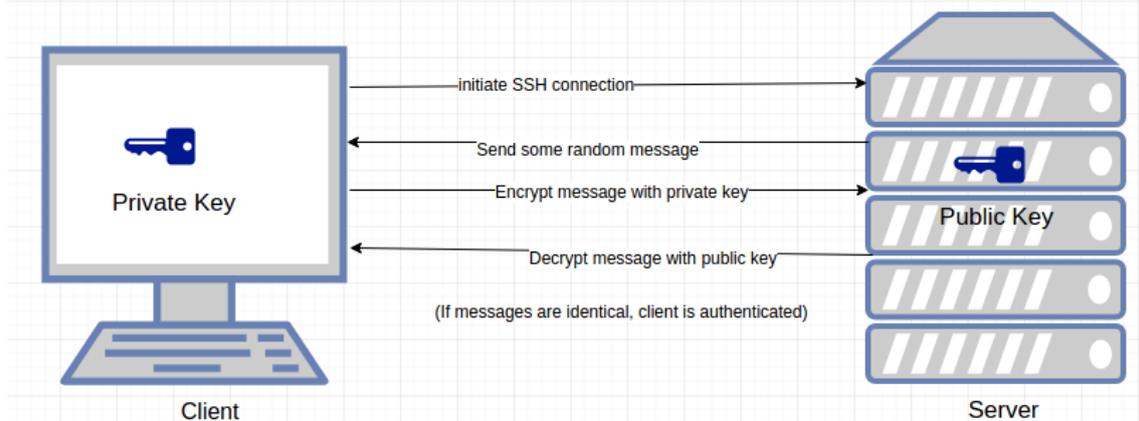


Figura 6.7: Diagrama de mensajes entre cliente y servidor para la autenticación de la conexión SSH [37]

6.3.4. Webmin

Webmin es un panel de control de código abierto que permitirá administrar y monitorizar de una forma más sencilla los sistemas. Su instalación es muy sencilla, se ha seguido la guía que publica webmin en su página web [39] para instalarlo.

Se ha elegido Webmin principalmente porque es gratuito y ofrece todas las herramientas necesarias para el control de los servidores. Además al ser de código abierto existen muchos «plugins» que añaden funciones útiles.

El paquete de Webmin no está incluido en el repositorio base de APT que ofrece Debian, es por esto que se deberá descargar el paquete .deb desde la página web oficial e instalarlo a mano usando el administrador de paquetes DPKG.

Existe otra forma de instalar Webmin que es a través de repositorios, para ello se deberá añadir el repositorio oficial de Webmin a la lista de repositorios del APT para lo que se necesitará una llave de «Gnu Privacy Guard»(GPG). Una vez instalado y actualizado el repositorio se podrá instalar Webmin como un paquete más de APT. [39] [40].

Tras la instalación el servidor estará disponible en "https://IP:10000", donde IP deberá sustituirse por la dirección IP de la máquina donde se haya instalado. Al acceder dará errores de dependencias, paquetes software que Webmin necesita pero no están instalados, estos paquetes se encuentran en la guía oficial anteriormente mencionada [39].

También es normal que se utilicen certificados SSL autofirmados para este tipo de servidores, ya que suelen ser accesibles solamente desde la red interna y no necesitan tanta seguridad, lo que provocará errores del navegador que en algunas ocasiones no permitirán el acceso a la web por no ser segura. Para estos casos en los que estamos en una red local, podemos solucionar este problema indicándole a Webmin que no use seguridad «SSL» y por tanto dejará de alojar la página web como "https://IP:10000" y la alojará como "http://IP:10000". Esto se podrá realizar si modificamos la variable "ssl=1" a "ssl=0" en "/etc/webmin/miniserv.conf" [41].

Con esto último no se tendrá ningún problema para acceder a la web y poder controlar los servidores de una forma sencilla.

6.3.5. Servidor Multimedia

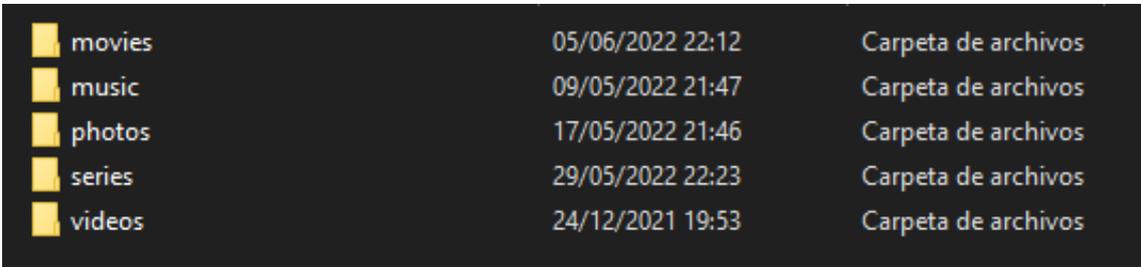
La instalación de Emby es muy parecida a la de Webmin vista en el apartado 6.3.4.

Primero se debe descargar el paquete .deb desde la web oficial, esto se puede hacer mediante una petición HTTP con el programa cURL. Una vez descargado se procede a instalarlo mediante el gestor de paquetes DPKG. Cuando el gestor haya acabado de instalarlo se debe revisar que el servidor este activo y que siempre que el NAS se encienda este también se tiene que encender, esto se logrará ejecutando los siguientes comandos:

```
sudo systemctl restart emby.service sudo systemctl enable
emby.service
```

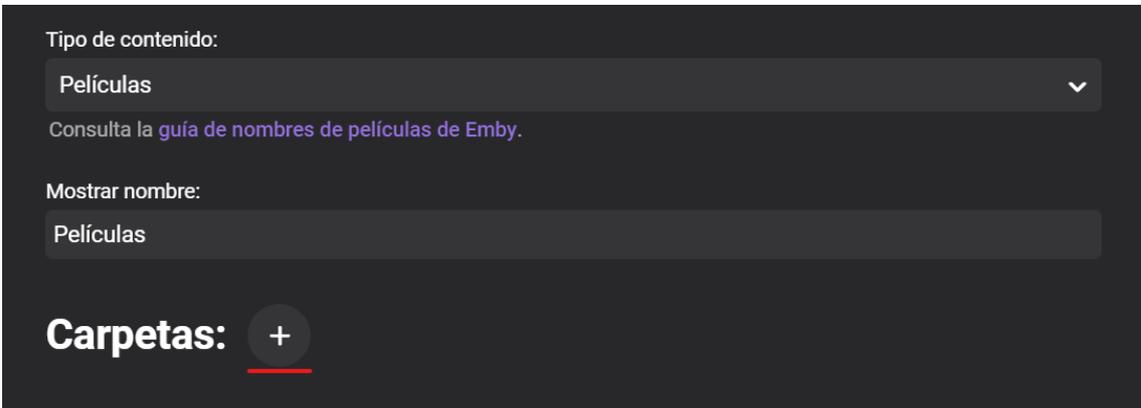
Ahora se debe configurar el servidor desde la web accediendo a “http://IP:8096”. Esta configuración es muy sencilla ya que el propio Emby indica que hacer en cada momento. Se va a utilizar la partición ya hecha en SMB, tal y como se indica en el apartado 6.3.2, de esta manera los recursos que muestre Emby podrán ser modificados en cualquier momento. En este directorio se crearán distintas carpetas para representar las distintas categorías de contenido, ya sea vídeos, música o fotografías, un ejemplo sería la figura 6.8 en la que se ven carpetas para: películas, música, fotografías, series, etc.

Una vez creadas las carpetas se debe indicar a Emby qué contendrá cada una de ellas como se ve en la figura 6.9. En esta misma figura 6.9 se ha marcado donde hay que pulsar para poder añadir la carpeta, se deberá poner el nombre completo del directorio, es decir, en este caso será «/media/share/multimedia/movies». Una vez añadidas todas las carpetas estas aparecerán en la página inicial de Emby y se podrá acceder a ellas para ver el contenido.



movies	05/06/2022 22:12	Carpeta de archivos
music	09/05/2022 21:47	Carpeta de archivos
photos	17/05/2022 21:46	Carpeta de archivos
series	29/05/2022 22:23	Carpeta de archivos
videos	24/12/2021 19:53	Carpeta de archivos

Figura 6.8: Ejemplo de estructura de directorios en la partición multimedia de SMB



Tipo de contenido:
Películas

Consulta la [guía de nombres de películas de Emby](#).

Mostrar nombre:
Películas

Carpetas: +

Figura 6.9: Interfaz Emby para seleccionar que tipo de contenido tendrá la carpeta

6.4 Configuración Wake On Lan

«Wake On Lan» es un protocolo que suele estar presente en la gran mayoría de ordenadores y que se puede activar desde la propia BIOS del PC. En este caso la opción se encuentra en el apartado de “Encendido del ordenador mediante PCIe”, un ejemplo de esta opción se ve en la figura 6.10. Este procedimiento se debe hacer en los ordenadores que se quieran despertar de forma remota.

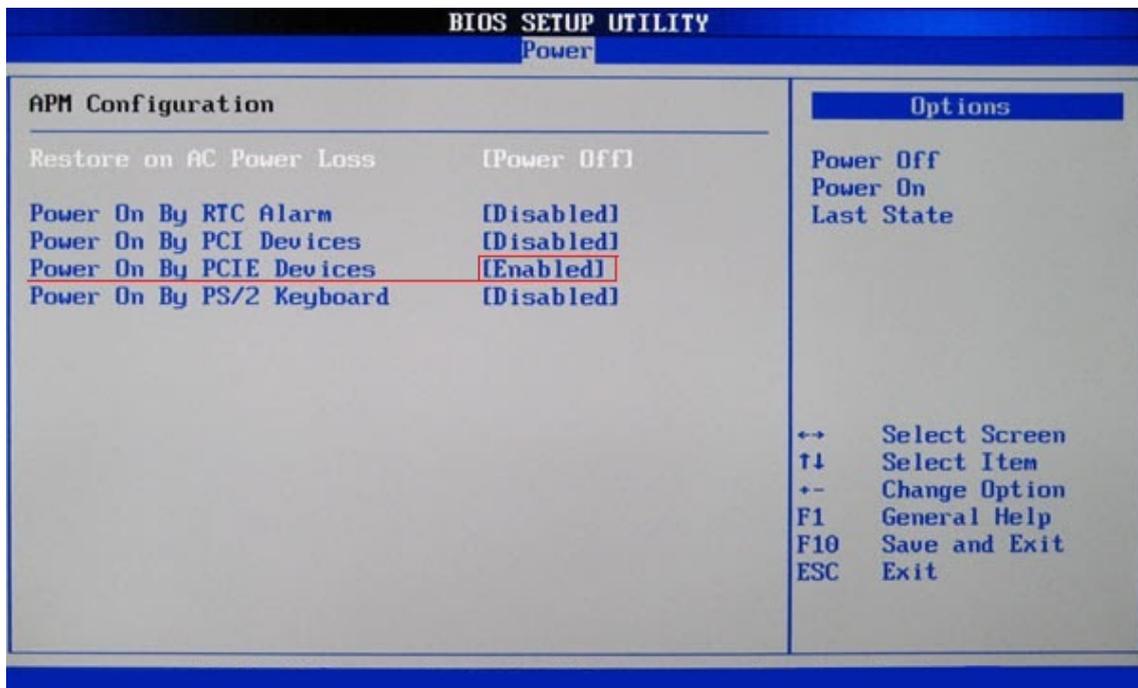


Figura 6.10: Interfaz de «BIOS» en la que se ve la opción para despertar el servidor mediante dispositivos conectados a la interfaz PCIe

Mencionar que la Raspberry PI, en el momento de redactar este documento, no permite el uso de «Wake On Lan» sobre ella misma para poder encenderse de forma remota, aunque no importa demasiado porque el consumo de este dispositivo es mínimo y realmente no merece la pena apagarlo si no es necesario.[42]

Para poder convertir la Raspberry PI en un servidor «Wake On Lan» es necesario cierto software, en este caso “wakeonlan”, así se podrá despertar a los otros ordenadores. Este programa se instala con el gestor de paquete APT.

El software permite enviar «Magic Packets» a la dirección MAC que se le indique como parámetro, encendiendo así los ordenadores que tengan activado el encendido por «Wake On Lan».

6.5 Creación de scripts

En este apartado se va mostrar y explicar como funciona el script para detectar que un usuario se ha conectado a la VPN de forma satisfactoria. Además también se explicará la necesidad y como funciona la página web para encender ordenadores.

Para que el NAS, el ordenador de sobremesa, se despierte cuando algún usuario entre a la VPN será necesario crear un «sniffer» de archivos de registro en la Raspberry PI, cuando esta detecte un usuario enviará un «Magic Packet» al NAS para encenderlo.

Para poder estar observando constantemente un archivo se ha utilizado un programa llamado «inotify». Este programa avisa cada vez que un archivo ha sido modificado y vuelve a la 'escucha', de esta forma cada vez que este programa devuelva un evento se puede obtener las últimas líneas del archivo de registros de la VPN y mediante filtros «grep» y expresiones regulares obtener las líneas de inicio de sesión, posteriormente se compara la línea obtenida con lo que se supone que devuelve la VPN cuando un usuario se conecta de forma satisfactoria. Si estos registros coinciden entonces el servidor NAS se despierta.

Para el caso de la desconexión es igual que antes, cuando el registro obtenido coincide con una desconexión se procederá a apagar el servidor, o suspenderlo. Este apagado no se hará de forma instantánea sino con histéresis, es decir, se esperará unos minutos antes de apagar el servidor por si otro usuario se llegara a conectar antes de apagarlo. La propia VPN es capaz de apagarse sola, y por ende el NAS, si detecta que el cliente está "muerto", es decir, que el cliente no contesta a los paquetes «keep-alive».

Por otro lado si se quiere despertar el servidor de forma manual también se ha creado una página web en «php» con «Apache» para poder hacerlo utilizando la misma tecnología de «WOL». En la figura 6.11 se muestra el aspecto de dicha página.

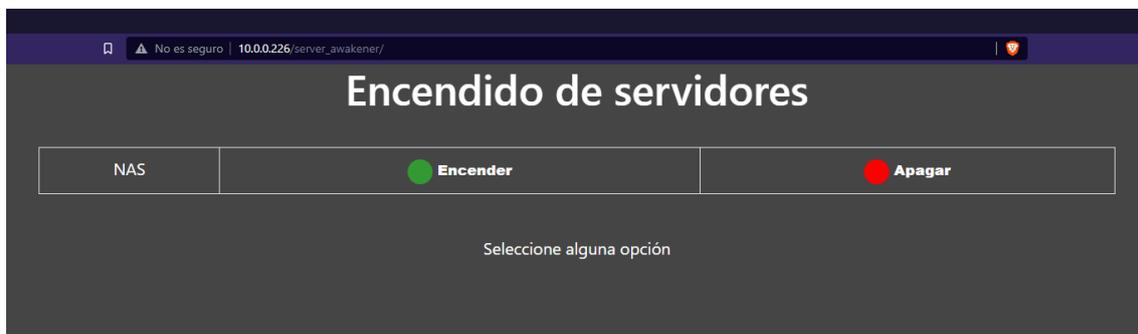


Figura 6.11: Web propia creada en PHP donde se ven los dos botones para encender o apagar el ordenador

Para obtener una explicación más en detalle revisar el apéndice A.6.

6.6 Configuración del router

La configuración del router por defecto cierra todos los puertos para evitar que elementos externos a la red accedan a ella. Esto es una medida de seguridad muy buena pero para este caso es un problema porque no permite la conexión de usuarios a la VPN. Es por esto que es necesario abrir los puertos que la VPN necesita, en este caso los puertos 4500 y 500 ambos por protocolo UDP, es decir solo acepta paquetes UDP y rechaza los TCP. El primer puerto, 4500, es un puerto para el «NAT-traversal» de IPsec, en cambio el 500 es un puerto destinado para el protocolo IKE, para el intercambio de llaves.

A parte de abrir los puertos también se hará «port-forwarding», explicado en el apartado 4.2, que se trata de redireccionar las peticiones que entran por cierto puerto público a un dispositivo de dentro de la red local. Como se ve en la figura 6.12 para realizar el «port-forwarding» es necesario especificar el dispositivo al que se quiere redireccionar las peticiones poniendo su dirección MAC, que en este caso se encuentra tapada, y también especificar que puerto público se quiere abrir y cual es el puerto del dispositivo a donde se quiere enviar los paquetes, puerto local.

De esta forma todos los puertos públicos están cerrados excepto dos que envían todo lo que reciben a un dispositivo en concreto.

Editar asignación de puertos

Nombre del servicio	<input type="text" value="VPN IPsec"/>
Dispositivo	<input type="text" value="██████████"/> ▾
LAN IP	<input type="text" value="192"/> . <input type="text" value="168"/> . <input type="text" value="0"/> . <input type="text" value="226"/>
Protocolo	<input type="text" value="UDP"/> ▾
Tipo	<input checked="" type="radio"/> Puerto <input type="radio"/> Intervalo de puertos
Puerto público	<input type="text" value="4500"/>
Puerto local	<input type="text" value="4500"/>
<input type="button" value="Guardar"/> <input type="button" value="Cancelar"/>	

Figura 6.12: Captura de la configuración del «Port Forwarding» del puerto 4500

6.7 Configuración del DDNS

Como se ha mencionado en el apartado 5.7 donde se explicaba la tecnología DDNS, para este proyecto se usará el servidor propio de IONOS al cual se puede acceder con una clave para el API de desarrolladores.

Para poder usar el servidor es necesario crear esta clave del API de IONOS ya que se necesitará para poder autenticarse en el API de ellos y poder hacer peticiones para obtener la información que se necesita. Se necesitará hacer una petición de tipo «POST» a la url “/v1/dyndns” tal y como se ve en la figura 6.13 donde se deberá sustituir “example-zone.de” por el dominio deseado.

Tras ejecutar la petición nos deberá devolver una respuesta «200 OK», si todo es correcto, con un objeto JSON que contendrá un url. Esta url será la encargada de actualizar la IP a la que apunta el servidor DDNS.

Esto se configurará añadiendo la url anterior como servidor propio de DDNS en la configuración del dominio desde el panel de control de IONOS. Por ultimo habrá que añadir un «cronjob», una tarea que se ejecuta cada cierto tiempo. En este caso se ejecutará cada hora y hará una petición con curl a la url antes mencionada, quedaría de la siguiente forma:

```
* */1 * * * curl https://ipv4.api.hosting.ionos.com/dns/v1/dyndns?
lista-de-chars-propios-de-cada-url
```

The screenshot shows the configuration for the 'Dynamic DNS' endpoint in the IONOS API. The endpoint is a POST request to '/v1/dyndns'. The description states: 'Activate Dynamic Dns for a bundle of (sub)domains. The url from response will be used to update the ips of the (sub)domains.' The 'Parameters' section is empty, with a 'Cancel' button. The 'Request body' is required and set to 'application/json'. An 'Examples' dropdown is set to '[Modified value]'. A text area contains the following JSON configuration:

```
{
  "domains": [
    "example-zone.de",
    "www.example-zone.de"
  ],
  "description": "My DynamicDns"
}
```

At the bottom, there are 'Execute' and 'Clear' buttons.

Figura 6.13: Captura del apartado «Dynamic DNS» de la API de IONOS

CAPÍTULO 7

Seguridad

La seguridad es un apartado muy importante en una VPN, si esta falla todos los datos de los usuarios pueden ser comprometidos y más teniendo en cuenta que esta VPN está abierta a internet, además si no se añade ninguna protección es una puerta abierta para cualquier atacante a la red doméstica.

Una de las prácticas más famosas y eficientes es la seguridad por capas, es decir, no enfocarse en un único punto de seguridad si no en todos, esto incluye la seguridad de la red, seguridad de dentro del sistema y seguridad en la propia VPN.

Los tres elementos más importantes de la seguridad son: la confidencialidad, la integridad y la disponibilidad. Si estos tres elementos se cumplen podemos decir que el sistema es seguro.

Es por esta razón que se quiere añadir ciertos elementos de seguridad que impidan y/o dificulten la entrada a la subred. Estos elementos son principalmente Fail2ban e IPtables.

Por otro lado también se van a desarrollar ciertos problemas de seguridad que surge en cuanto al intercambio de llaves Diffie-Hellman en dispositivos Windows y como solucionarlo.

Además se explicará la importancia de los permisos de usuarios UNIX y como se utilizan en este proyecto.

7.1 VPN

7.1.1. Problema seguridad IKE (modp1024)

La seguridad de la VPN se basa en dos elementos esenciales, el establecimiento de conexión (obtener llaves simétricas) y el cifrado a través de las llaves simétricas. El establecimiento de la conexión usa el protocolo Diffie-Hellman para poder crear estas llaves simétricas de forma confidencial, la resistencia de este protocolo depende en gran medida en el tamaño de las llaves que se usan, este tamaño está determinado por el «modp» cuanto más grande sea este valor de bits mayor seguridad ofrece el protocolo [44].

El tamaño mínimo recomendado por strongSwan es de dos kilobits, es decir «modp2048». El problema surge cuando se quiere utilizar la VPN en sistemas Windows y este no permite el uso de «modp2048» sino «modp1024», un tamaño de un solo kilobit. Esto supone un gran problema porque si se consiguiera romper este protocolo toda la seguridad de la VPN se vería comprometida, ya que esta es tan fuerte como su eslabón más débil [45].

Por último indicar que este error es conocido por la comunidad y realmente Windows 10 acepta tamaños de claves más grandes (como «modp2048») pero este requiere que se

active expresamente. Esta activación requiere modificar los registros del sistema operativo por tanto se explicará en el apartado 7.1.2.

7.1.2. Solución Windows 10

Modificar registros del sistema en Windows 10 siempre es un poco «peligroso» ya que si se llegara a modificar registros importantes se puede corromper el sistema operativo.

Este registro que se modificará es:

```
HKEY_LOCAL_MACHINE/System/CurrentControlSet/Services/Rasman/  
Parameters/NegotiateDH2048_AES256
```

Para poder acceder a esta dirección se usará el editor de registros de Windows. Se buscará la dirección en el buscador de arriba y se modificará el archivo «NegotiateDH2048_AES256». Si este registro no existe se añade uno nuevo como una «DWORD» con el mismo nombre. Una vez se pueda acceder al registro se debe modificar el valor que tenga por un «1» o por un «2», siendo «1» para aceptar «modp2048» y «2» para obligar a usar «modp2048». Esta modificación se puede ver en la figura 7.1

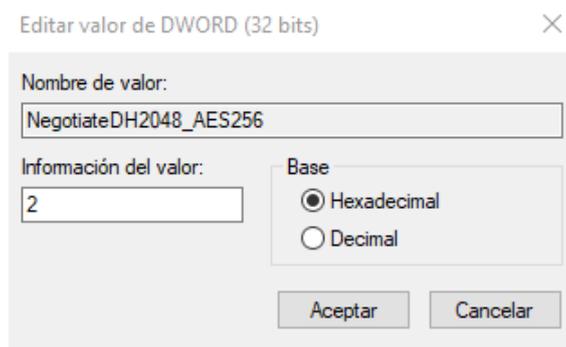


Figura 7.1: Captura del registro modificado

Esta modificación permite poder aceptar grupos mayores de 1024bits pero viene con una pega, si se aceptan estos grupos no se puede aceptar cifrados más eficientes como «aes256gcm16» como se vio en el apartado 6.3.1 y el sistema solo acepta «aes256 Cipher block chaining (CBC)», un algoritmo muy seguro pero no tan eficiente como el anterior[49]. En este caso se ha decidido mantener el «modp2048» en vez del «aes256gcm16».

7.2 Firewall

El cortafuegos o «Firewall» es un elemento de mucha importancia en esta red, ya que al tener puertos expuestos a Internet hay que asegurarse de que todo lo que entra por esos puertos es lo que queremos, en este caso paquetes UDP. Como se ve en la figura 4.1 el cortafuegos se situará entre la puerta de enlace y el servidor VPN, de esta forma se podrá filtrar los mensajes de una forma mucho más sencilla.

La herramienta que se usará para crear este «firewall» es IPtables, con este software se podrá indicar que paquetes pueden pasar, por donde pueden pasar o incluso que paquetes bloquear directamente. Gracias a esto se pueden detectar patrones de ataque, como el ataque SYN-ACK en TCP [46], y bloquearlos antes de que puedan causar ningún daño.

Para este caso vamos a hacerlo sencillo y solo se añadirán las reglas necesarias para el correcto funcionamiento de la VPN y páginas web alojadas en el sistema.

Para la VPN se necesitará permitir el acceso por los puertos 4500 y 500 a los paquetes UDP. En cambio para la página web se necesitará solamente el puerto 80, ya que no se usa HTTPS, a los paquetes TCP. Al resto de puertos se les indicarán que descarten («drop») o que no hagan caso de los paquetes que reciban. Importante indicar que se debe hacer «DROP» a los paquetes y no «REJECT» porque este último devuelve un mensaje de error como respuesta dando la información de que el dispositivo se encuentra en esa dirección IP, en cambio «DROP» no devuelve nada y por tanto no se puede saber si realmente el dispositivo está ahí.

Para saber más sobre las reglas de IPtables visitar el apéndice B.1.

7.3 Fail2ban

Fail2ban o «Fail to ban» es un «sniffer» de archivos de registro que se usará para bloquear direcciones IP que intenten acceder a la VPN y fallen en el intento.

En un principio la idea era utilizar un «script» propio que funcionara de manera similar a fail2ban usando expresiones regulares propias para filtrar los registros, pero debido a que el bloqueo de IPs sería manual y las expresiones tenían cierta complejidad, se decidió usar este software principalmente para no reinventar la rueda.

Como ya se ha explicado en el apartado 7.2 el «firewall» permite filtrar de una manera rudimentaria los paquetes que recibe un sistema pero, ¿y si se produce un ataque de fuerza bruta en la VPN para adivinar la contraseña?, el «firewall» no sirve porque este no revisa el contenido de los paquetes. Aquí es donde entra fail2ban.

La principal función de fail2ban es proteger contra algunos ataques de denegación de servicio y ataques de fuerza bruta que intenten averiguar la contraseña mediante prueba y error.

Este software es interesante por que permite integrar distintos filtros o expresiones regulares que sirven para muchos protocolos y gracias a la comunidad que hay detrás se pueden usar filtros ya creados para la mayoría de protocolos.

En este caso usaremos un filtro ya creado para «strongSwan IKEv2» que se encuentra en «GitHub» del usuario «s1nnerman89»[33]

Este filtro bloqueará permanentemente cualquier dirección IP que haga tres intentos fallidos de conexión. Se puede modificar el número de intentos y cuanto tiempo estarán bloqueados pero para este caso lo dejaremos tal y como viene.

Para usar estos filtros el procedimiento es muy sencillo. Solamente hay que crear dos archivos, uno “NOMBRE.conf” y otro “NOMBRE.local” en “/etc/fail2ban/filter.d” y “/etc/fail2ban/jail.d” respectivamente.

Una vez creados estos archivos reiniciamos el proceso para que se apliquen los cambios.

7.4 Permisos UNIX, SAMBA y SSH

Los permisos de usuario son el elemento de seguridad más importante del sistema porque limitan en gran medida las funciones que se pueden realizar dentro de él mismo. Siempre es buena práctica usar el usuario root lo mínimo posible ya que este tiene control total sobre el dispositivo, lo mejor es seccionar y atomizar el poder de cada usuario, es

decir, especializar a cada usuario en una tarea. Por ejemplo para el control de la página web se usa un usuario llamado apache que solo tiene permisos en los directorios donde se aloja la web y sobre el servidor web. Otro ejemplo es el usuario "wakeonlan" que simplemente se utiliza para ejecutar los scripts que despiertan al servidor NAS, de esta forma evitamos el escalado de poder en el sistema.

Aparte de usuarios especializados en ejecutar procesos también existen los usuarios personales, estos usuarios deben tener limitadas sus funciones y solo pueden trabajar en los directorios que se permitan, por ejemplo en sus directorios "home". Gracias a estas limitaciones aseguramos que estos usuarios no malogren el sistema y se pueda mantener la confidencialidad.

Para asegurar el acceso por SSH se ha creado un grupo denominado "admin". Dentro de este grupo se podrán añadir usuarios y estos tendrán permisos "root" y acceso por SSH, esta limitación de acceso se ha conseguido configurando los archivos de OpenSSH y añadiendo una regla de excepción para los usuarios que pertenezcan a este grupo, ver apéndice A.5

Para el apartado de SAMBA se debe tener en cuenta que los archivos que estos usuarios creen también deben de tener permisos y estos deben ser los adecuados para mantener la confidencialidad.

Si se quiere saber más sobre la configuración de los permisos en SAMBA visitar el apéndice A.4.

CAPÍTULO 8

Pruebas

8.1 Rendimiento LAN

Para las pruebas dentro de la red local se va a utilizar una herramienta denominada ChowEazyCopy creada por el usuario “Cinchoo” de GitHub [48]. Este software permite enviar archivos a otros discos físicos o en red.

Se realizarán diez envíos de cinco ficheros con tamaños distintos que comprenderán desde 1KiB hasta 10GiB de peso. De los diez envíos por archivo se obtendrá la velocidad media y se verá como se comporta la red.

Para saber las características de la red visitar el apartado 4.2. Se transferirán los archivos desde un NVME M.2 WD Black hasta el disco HDD en remoto de la NAS a través de una red de un 1Gb/s y por protocolo SMB.

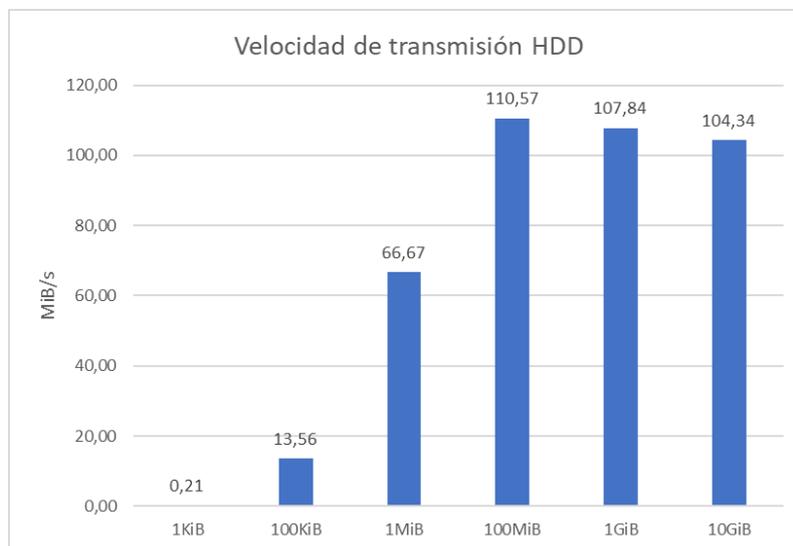


Figura 8.1: Gráfica que representa la velocidad media de transferencia de un archivo variando su tamaño en un disco HDD

El resultado de la prueba se muestra en el gráfico de la figura 8.1 y tal y como se ve la transferencia de archivos pequeños es muy poco eficiente ya que si se transfiere mil archivos de 1KiB, 1MiB en total, sin comprimir, la velocidad de transferencia sería muy similar a la velocidad del archivo de 1KiB, tal y como se ve en la figura 8.2, por esta razón es tan importante comprimir los archivos ya que una vez comprimidos la velocidad de transferencia sería igual que la del archivo de 1MiB. También se observa que cuanto más

```

-----
Director.:      Total   Copiado   OmitidoNo coincidencia   ERROR   Extras
Archivos:      1001     1000     1         0         0         0
Bytes: 10.000 g 1000.0 k 10.000 g   0         0         0
Tiempo: 0:00:06 0:00:06           0:00:00 0:00:00

Velocidad:      152267 Bytes/s
Velocidad:      8.712 Megabytes/min
Finalizado: jueves, 30 de junio de 2022 22:51:14

```

```
>exit
```

Figura 8.2: Captura del resultado de transferir 1000 archivos de 1KiB sin comprimir

aumenta el tamaño de los archivos mayor es la velocidad siendo el archivo de 100MiB el que mayor velocidad alcanza, a partir de aquí la velocidad disminuye (posiblemente por culpa del HDD).

Para probar que el disco HDD esté siendo un cuello de botella se han hecho las mismas pruebas pero esta vez la carpeta compartida está ubicada en una SSD instalada en el NAS.

Tal y como se ve en la figura 8.3, los datos no varían mucho aunque el SSD sea casi cuatro veces más rápido en lectura y escritura, 400MB/s y 200MB/s respectivamente, que el HDD. Sí que es verdad que los archivos más grandes no disminuyen en velocidad pero teniendo en cuenta que la transmisión de archivos tan grandes se va a hacer de forma muy ocasional es preferible tener más espacio, 1 TiB, que la velocidad extra que proporciona el SSD.

Queda demostrado de esta manera que la red de 1Gb/s que es equivalente a 125MB/s en teoría, unos 110 MB/s prácticos, queda saturada y por tanto se saca el máximo rendimiento de la misma.

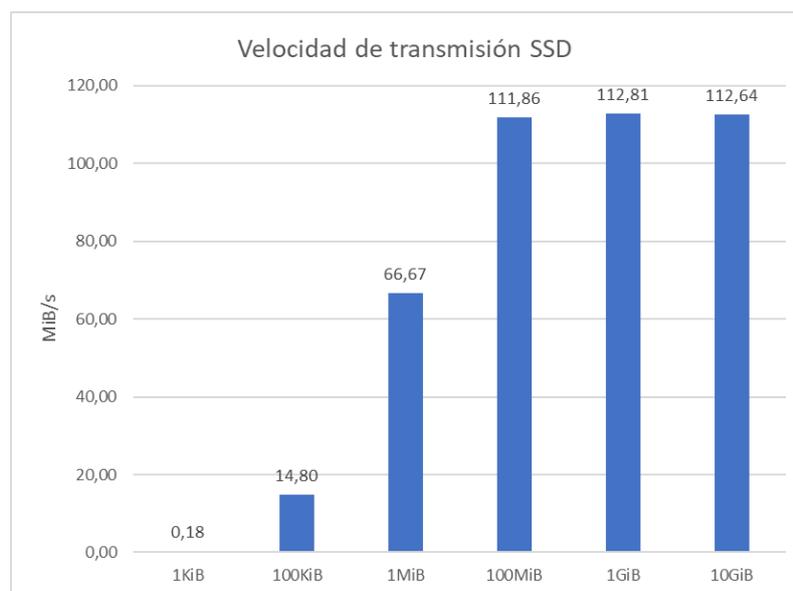


Figura 8.3: Gráfica que representa la velocidad media de transferencia de un archivo variando su tamaño en un disco SSD

8.2 Rendimiento VPN

El escenario para esta prueba es igual que en el anterior apartado con el HDD pero con un cuello de botella importante que es la velocidad de subida que se convierte en la velocidad de bajada del usuario que utiliza la VPN. Por tanto el límite teórico que existe es de 100Mb/s, unos 12,5 MB/s, a partir de aquí podemos ver el rendimiento de la VPN y que porcentaje se pierde por el «overhead» de esta. Es importante mencionar que para eliminar posibles problemas con la red esta prueba se ha hecho varias veces en días distintos y a distintas horas, en total se han hecho 120 transferencias de archivos.

Como se ve en la figura 8.4 la velocidad más alta es de 7,38 MiB/s, 60Mb/s aproximadamente, teniendo en cuenta que el límite es de 100Mb/s se está perdiendo un 40 % de ancho de banda. Esta pérdida de rendimiento se puede deber a varios factores, ya sean los algoritmos utilizados en la VPN o simplemente las propias cabeceras añadidas por la VPN («overhead»).

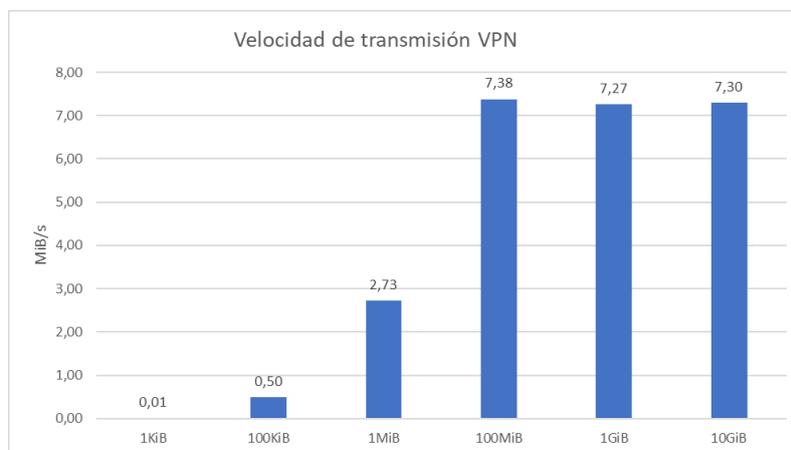


Figura 8.4: Gráfica que representa la velocidad media de transferencia de un archivo variando su tamaño con la VPN

Como se vio en el apartado anterior 6.3.1 el algoritmo seleccionado, ver figura 8.5, es el AES-256-GCM-16» (se ha utilizado dispositivo Linux ver apartado 7.1.2) que es extremadamente rápido, por tanto el rendimiento perdido por parte de este es poco. El tráfico de la red está descartado por lo mencionado anteriormente y por último y más probable las cabeceras. Tal y como se aclara en el apartado 5.4 para que una VPN funcione debe encapsular el contenido que se quiere transmitir en otro paquete, en este caso en una trama UDP, estas cabeceras extra aumentan el tamaño de la trama pero esta solo puede tener un máximo de 1518B por trama, esto significa que hay que quitar bytes de otro lugar y ese sitio son los datos transmitidos, por esta razón el ancho de banda baja porque los datos útiles por trama disminuyen.

```
109[CFG] <8> selected proposal: IKE:AES_GCM_16_256/PRF_HMAC_SHA2_384/MODP_2048
109[CFG] <8> received supported signature hash algorithms: sha256 sha384 sha512 identity
109[IKE] <8> local host is behind NAT, sending keep alives
109[IKE] <8> remote host is behind NAT
109[IKE] <8> DH group ECP_256 unacceptable, requesting MODP_2048
```

Figura 8.5: Registro donde indica el algoritmo seleccionado por el usuario

En el siguiente apartado, 9.1, se verá que el sistema aún presenta limitaciones que no son causadas por el hardware sino por el software, en este caso se trata de la carencia de «multithreading» del flujo de datos o «datastream» de la VPN. Esto significa que aunque se

disponga de cuatro núcleos de «CPU» el tratamiento de los datos que se envían y reciben por la VPN solo lo realiza un núcleo del procesador, tal y como se ve en la siguiente figura 8.6 donde dos usuarios de VPN («cloud2» y «cloud») distintos están haciendo una prueba de velocidad de internet a través de la VPN a la vez y en la parte de abajo se ve la carga del procesador del servidor VPN con «htop». Esta carga refleja que la VPN solo está usando un núcleo de los cuatro que tiene disponibles.

```

ubuntu@cloud2:~$ speedtest-cli
Retrieving speedtest.net configuration...
Testing from Vodafone Ono ( )...
Retrieving speedtest.net server list...
Selecting best server based on ping...
Hosted by OLIVENET (Málaga) [467.34 km]: 66.945 ms
Testing download speed.....|

ubuntu@cloud:~$ speedtest-cli
Retrieving speedtest.net configuration...
Testing from Vodafone Ono ( )...
Retrieving speedtest.net server list...
Selecting best server based on ping...
Hosted by OLIVENET (Málaga) [467.34 km]: 66.364 ms
Testing download speed.....|

1 [|||||||||||||100.0%] Tasks: 58, 162 thr; 3 running
2 [||| 1.3%] Load average: 0.20 0.26 0.21
3 [||| 1.3%] Uptime: 13 days, 00:47:39
4 [ 0.0%]
Mem[||||| 287M/3.75G]
Swp[ 0K/1024M]

```

Figura 8.6: Test de dos usuarios concurrentes usando la VPN donde se ve que solo se está usando un núcleo de la CPU del servidor VPN.

8.3 Funcionamiento Scripts

Para comprobar que el «sniffer» funciona correctamente compararemos los archivos de registro de la VPN con los archivos de registro del propio «script» para ver si las horas coinciden y el «script» detecta correctamente el inicio de sesión.

Comparando las figuras 8.7 y 8.8 podemos ver que la fecha y hora entre los dos registros es la misma y por tanto el «script» detecta correctamente el inicio de sesión del usuario.

Que el «IKE_SA» se haya establecido de forma satisfactoria asegura que el usuario ha hecho correctamente la autenticación EAP y que este dispone del certificado adecuado.

```

Jul 20 21:51:27 28[MGR] <ikev2-vpn|9> checkin IKE_SA ikev2-vpn[9]
Jul 20 21:51:27 28[MGR] <ikev2-vpn|9> checkin of IKE_SA successful

```

Figura 8.7: Registros donde se indica que la creación del IKE-SA ha sido correcta, esto indica que el usuario conectado ha superado el EAP y tiene el certificado correcto.

```

[LOG] [20-07-2022 21:30:01] ##### Inicio Script #####
[LOG] [20-07-2022 21:51:27] Jul 20 21:51:27 19[MGR] <9> checkin of IKE_SA successful
Sending magic packet to 255.255.255.255:9 with ██████████
[LOG] [20-07-2022 21:51:28] ██████████ awaked with IKE_SA successful
[LOG] [20-07-2022 21:51:29] User: '██████████' has just connected

```

Figura 8.8: Registros donde se indica que el «sniffer» ha detectado correctamente que la creación del IKE-SA ha sido correcta. La información censurada son direcciones MAC y el nombre del usuario.

8.4 Cifrado

En esta sección de prueba revisaremos el contenido de los paquetes enviados desde la VPN para ver si realmente estos se están cifrando correctamente. Para poder visualizar el contenido se utilizará la herramienta «tcpdump». Para la prueba se utilizará el protocolo «ICMP» con datos de ejemplo.

Para filtrar los paquetes ICMP se ha usado el siguiente comando:

```
sudo tcpdump -i enp0s3 -A icmp
```

En cambio para los paquetes cifrados el comando es más difícil ya que no se puede ver si es un paquete ICMP por tanto se va a buscar datagramas UDP que se dirijan al puerto 4500, además es importante mencionar que se debe observar la interfaz física (enp0s3) y no la virtual creada por la VPN (ipsec0) ya que si no el tráfico de esta última interfaz se vería sin cifrar:

```
sudo tcpdump -i enp0s3 -A "udp port 4500"
```

El resultado del paquete ICMP cifrado es el que se ve en la figura 8.9. El paquete sin cifrar se puede ver en la figura 8.10.

```
23:16:32.292014 IP 10.10.10.169.50953 > : UDP-encap: ESP spi=..., seq=0x39f, length 104
E...y.@...
.Q..s. ...p;s...(.4.[...9..z...\.11u-...oX ..N.%(ZG.....g.q...M.*.6!...g....\..a..15.....<3A.....z.pQV../G.m
```

Figura 8.9: Datagrama encapsulado en UDP de un paquete ICMP cifrado.

```
23:19:16.774701 IP : ICMP echo reply, id 15, seq 2, length 64
E..Tb...7...Q..s
.....b.....a.....! "#$%&'()*+,-./01234567
```

Figura 8.10: Paquete ICMP sin cifrar en el que se ve su contenido.

Limitaciones y posibles mejoras

Las limitaciones que se van a exponer son problemas que disminuyen el rendimiento del sistema. De momento estas limitaciones no tienen una solución sencilla o no se pueden resolver en poco tiempo.

Para cada limitación también se expondrá una posible solución.

9.1 Multithreading

Como se explica al final del anterior apartado 8.2, cuando ya se ha establecido una conexión VPN los flujos de datos de esta, «datastreams», tienen que pasar por el servidor VPN en forma de paquetes IPsec. El tratamiento de estos paquetes se realiza en un solo núcleo del procesador, aunque haya diversos usuarios conectados y enviando paquete de información, esto es un problema porque limita el rendimiento de la VPN disminuyendo su ancho de banda, esto sucede porque los recursos de ese único núcleo se tienen que repartir entre el número de usuarios que estén conectados.

Esto sucede principalmente por una limitación del «kernel» de strongSwan, el cual dedica un hilo de procesamiento solo a cifrado y por defecto este hilo tiene asignado un núcleo del procesador [50]. A parte de la razón anterior también se ha de tener en cuenta que los paquetes tienen un cierto orden y estos se tienen que enviar tal y como han llegado, poner medidas para evitar condiciones de carrera si fuera paralelo provocaría un rendimiento peor al secuencial [51].

9.1.1. Posible solución

La solución que se va a proponer no es una paralelización como tal del proceso de cifrado, sino la distribución de usuarios a lo largo de todos los núcleos del procesador, para ello se usaría Docker y un proxy inverso como Nginx. Docker es una tecnología que permite separar los procesos del sistema usando contenedores, estos contenedores aíslan los procesos y por tanto se puede tener varias instancias de ellos. Teniendo varias instancias de una misma VPN, una por núcleo, se podría repartir cada usuario entre estas instancias con un proxy inverso que también actúe como balanceador de carga en modo Round-Robin. Para entender mejor la solución revisar el diagrama 9.1

Un ejemplo de funcionamiento sería el siguiente: se supone que se dispone de cuatro contenedores con el proceso VPN, con su propia dirección IP privada, estos contenedores se conectan a un proxy inverso de Nginx que reparte las conexiones de VPN, que recibe por la IP pública del servidor, a cada contenedor en modo Round-Robin. Dicho esto si un usuario se conecta Nginx redirigirá esta petición al primer contenedor (VPN 1) si después

se conecta otro este se redirigirá al segundo (VPN 2) y así sucesivamente. Con una correcta configuración de docker se puede destinar un núcleo de la CPU por contenedor y de esta manera se utilizaría el máximo del procesador.

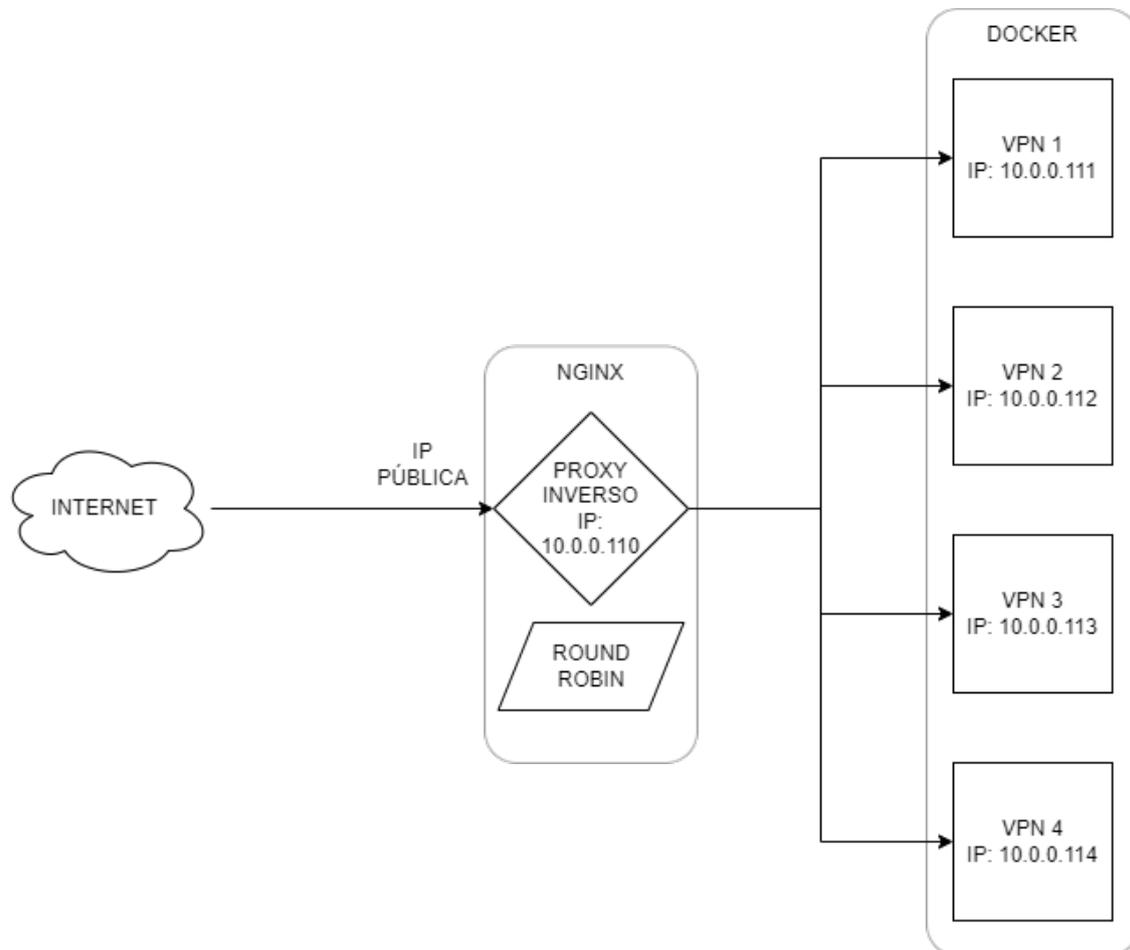


Figura 9.1: Diagrama de procesos Docker y proxy inverso.

9.2 Ancho de banda de internet

Para poder interactuar con internet es necesario dos tipos de ancho de banda, de bajada como de subida. El primero es el que limita la velocidad de descarga máxima de un objeto en internet, el segundo es la velocidad a la que se puede subir un objeto local a internet. Normalmente la velocidad de subida suele ser bastante menor a la de bajada.

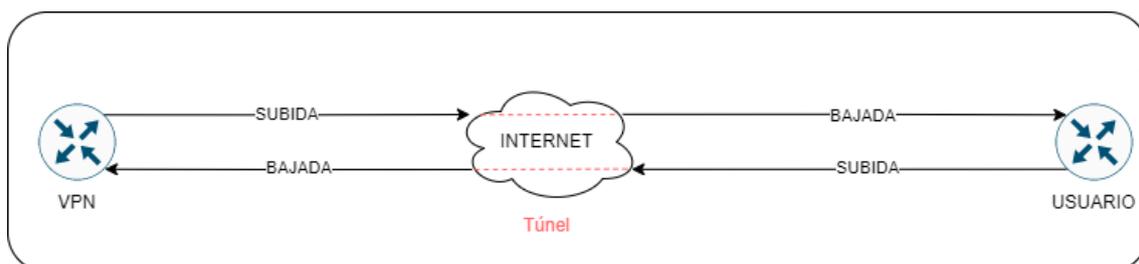


Figura 9.2: Diagrama de conexión entre el servidor VPN y un usuario en el que se ve que la velocidad de subida de la VPN es la velocidad de bajada del usuario .

Como se ve en la figura 9.2 en el momento en el que el túnel entre cliente y servidor se crea la velocidad de bajada del cliente depende de la velocidad de subida del servidor, esto sucede porque este último le tiene que “enviar” los datos al cliente a través de internet (subir un objeto a internet). Por esta razón existe un gran cuello de botella en la velocidad de descarga del cliente ya que depende directamente de la velocidad de subida del servidor.

La única solución posible en este caso es contratar más velocidad de subida al proveedor de internet si es posible.

CAPÍTULO 10

Conclusiones

Como conclusión se va a comprobar si los objetivos previstos han sido cumplidos, que objetivos aparecieron en el transcurso del proyecto y por último cuales fueron sus problemas y soluciones.

El principal objetivo es el estudio e instalación de una VPN y un servidor NAS. Este estudio se realizó con las tecnologías actuales más importantes de VPN, dando como resultado una VPN muy rápida y segura utilizando IPsec con IKEv2 y autenticación EAP. La instalación de esta VPN no fue sencilla y surgieron varios problemas, entre ellos la falta de compatibilidad con los protocolos de cifrado deseados que se instalaban por defecto, la solución fue buscar en la propia documentación específica del software de la VPN y encontrar parámetros avanzados que se ajustaran en las necesidades del proyecto. Otro problema fue la compatibilidad con los algoritmos utilizados en clientes Windows, que se pudo solucionar contactando con el soporte de Windows y buscando problemas similares en foros para ver posibles soluciones. Por otro lado al servidor NAS se le instaló el software SAMBA para compartir los archivos, el cual fue probado y dio como resultado que el cuello de botella de la velocidad de transferencia se origina en la propia red doméstica lo que significa que los servidores no reducen el rendimiento del sistema. El mayor problema que surgió con este servidor fue que no permitía que ningún usuario se conectara al archivo compartido, esto fue porque en la guía que se usó para esta instalación se omitió el paso de añadir el usuario a la base de datos del propio SAMBA y por tanto este usuario no “existía” para este programa. Por último el servidor de «streaming» multimedia que se instaló en el servidor NAS fue como primera opción «PlexMediaServer», ya que disponía de una interfaz web atractiva, buena comunidad para resolver problemas y era gratuito, pero la gran pega era que no admitía de forma nativa vídeos MKV y es por esto que se eligió finalmente Emby como servidor multimedia, ya que cumplía con todas las características anteriores de Plex y aceptaba vídeos MKV sin la necesidad de codificarlos en el momento para transmitirlos, aunque esto último depende del dispositivo del usuario que vea el contenido.

Por otro lado la configuración de un servidor DDNS también era un objetivo importante, ya que permite mantener una dirección estática aunque la dirección IP sea dinámica. En un primer momento se pensaba utilizar «duckdns.org» pero conforme se desarrollaba el proyecto se investigó una opción de DDNS con IONOS, ya que se tenía un dominio comprado con ellos, y finalmente se instaló usando su propia API y añadiendo un comando de IONOS a la propia Raspberry que se ejecutará de forma regular para actualizar los datos del servidor DDNS si fuera necesario.

Otro de los objetivos era la instalación de un panel de control para poder mantener y controlar mejor los servidores. El panel elegido fue «Webmin», la razones fueron dos, era gratuito y ya se tenía contacto con él por una asignatura de la carrera. En esta instalación

no hubo problemas mayores exceptuado que la web se iniciaba en modo «ssl», problema que se resolvió fácilmente buscando en la propia documentación del panel y modificando los archivos de configuración que se indicaban.

Por último se encuentra la creación de «scripts» y una página web propia para poder apagar o encender el NAS cuando se necesite. En la creación del «script» surgieron varios problemas, entre ellos como detectar modificaciones en archivos o como programar un proceso de apagado. Ambos problemas se resolvieron buscando soluciones propuestas para problemas parecidos y adaptándolas a las necesidades del proyecto. Además la propia página web al estar hecha en «php», lenguaje que se ha tenido que aprender, daba problemas para ejecutar comandos en usuario «root» y se ha solucionado creando un programa en C que ejecute el comando, de esta forma al ser un programa compilado no se puede modificar y se evita la escalada de permisos.

Por otro lado mientras se desarrollaba el proyecto se han añadido varios objetivos como la creación de un cortafuegos para el sistema o la instalación de «fail2ban» para evitar accesos no deseados a la red. Tras una formación en ambas tecnologías se cumplieron los objetivos sin ningún problema.

Después de realizar todo el proyecto la cantidad de tecnologías y métodos de estudio aprendidos son bastantes, se ha aprendido sobre tecnologías y protocolos de VPN, sobre algoritmos de cifrado e integridad y cual es su función e importancia, también sobre el uso de API y servidores DDNS. Se ha aprendido sobre el protocolo SMB y las buenas prácticas sobre el tratado de usuarios UNIX. Por otro lado se ha estudiado sobre la codificación de vídeos y el hardware necesario para ello, la necesidad de asegurar el sistema contra amenazas externas usando cortafuegos o programas, además se ha aprendido a desarrollar código tanto en «BASH» como en «PHP», por otro lado se han aprendido nociones básicas de mantenimiento de servidores en Linux tanto por consola como por el panel de control. Por último se ha investigado sobre otras tecnologías como Docker o Nginx.

Como conclusión decir que todos los objetivos principales han sido cumplidos y que gracias a este proyecto se han podido afianzar los conocimientos involucrados que se han enseñado en la carrera.

CAPÍTULO 11

Relación del trabajo desarrollado con los estudios cursados

Este trabajo surgió tras cursar una asignatura de cuarto llamada «Redes Corporativas». En esta asignatura aprendí exactamente lo que es una VPN y cual es su funcionalidad y tras comprar una Raspberry PI quise poner a prueba el conocimiento de esa materia. Otras asignaturas influyentes fueron «Redes» y «Fundamentos de sistemas operativos» de segundo, «Administración de sistemas», «Desarrollo Web» y «Diseño y configuración de redes de área local» de tercero y por último «Seguridad en redes y sistemas informáticos» en cuarto.

«Redes» sirvió principalmente como base para poder trabajar con paquetes TCP y datagramas UDP, además también para entender la base de ciertos protocolos. «Fundamentos de sistemas operativos» me dio la base para poder trabajar en Linux.

«Administración de sistemas» me enseñó qué es SAMBA, como tratar a los usuarios UNIX, que permisos otorgar y como otorgarlos. «Desarrollo Web» me explicó las bases del diseño de una web en «HTML» y con un poco de esfuerzo pude aprender «php» siendo autodidacta. Por último «Diseño y configuración de redes de área local» me enseñó las bases de una red local, cuales son los dispositivos necesarios y como poder gestionar esta red.

Por último «Seguridad en redes y sistemas informáticos» me permitió entender cual es la metodología adecuada para asegurar un sistema y que se considera seguro y que no.

Además este proyecto me ha ayudado a mejorar ciertas competencias transversales sobretodo «Planificación y gestión del tiempo», «Comprensión e integración» y «Análisis y resolución de problemas».

Bibliografía

- [1] Estadística sobre el uso de la nube en Europa en 2020 y 2021. Consultado en https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Cloud_computing_-_statistics_on_the_use_by_enterprises.
- [2] Protocolos de comunicación VPN más utilizados. Consultado en <https://ostec.blog/es/acceso-remoto/protocolos-comunicacion-vpn/>.
- [3] Uso de una Raspberry PI como servidor "Wake On Lan". Consultado en <https://notenoughtech.com/raspberry-pi/use-raspberry-pi-wol/>.
- [4] Vico Burruezo, M. (2021). Configuración de un servidor VPN alojado en una Raspberry. *Universitat Politècnica de València*. <http://hdl.handle.net/10251/172954>
- [5] Calderón Rodríguez, C. (2001). Implementación de una VPN (Virtual Private Network) usando el estándar IPSEC. *Universitat Politècnica de València*. <http://hdl.handle.net/10251/32198>.
- [6] Los ciberataques en España crecen un 125%. Consultado en https://cincodias.elpais.com/cincodias/2021/03/25/pyme/1616706362_846686.html.
- [7] Un incendio destruye parte del centro de datos de OVH en Estrasburgo, uno de los servidores más importantes de Europa. Consultado en <https://www.20minutos.es/noticia/4612250/0/>.
- [8] Redes Multigigabit NBASE-T: Características técnicas y equipos compatibles. Consultado en <https://www.redeszone.net/tutoriales/redes-cable/redes-multigigabit-nbase-t-caracteristicas-equipos/>.
- [9] Secure Shell. Consultado en <https://www.rfc-editor.org/rfc/rfc4253>.
- [10] SSH Handshake Explained. Consultado en <https://goteleport.com/blog/ssh-handshake-explained/>.
- [11] RSA (cryptosystem). Consultado en <https://www.rfc-editor.org/rfc/rfc3447>.
- [12] Samba Security Documentation. Consultado en https://wiki.samba.org/index.php/Samba_Security_Documentation#Modern_crypto.
- [13] Introduction to Server Message Block (SMB). Consultado en <https://www.educba.com/what-is-smb/>.
- [14] CRACKING PPTP VPNS. Consultado en <https://crack.sh/pptp/>.
- [15] Lukas Osswald, Marco Haerberle, and Michael Ment Performance Comparison of VPN Solutions. Consultado en <https://core.ac.uk/download/pdf/322886318.pdf>.

- [16] Performance Comparison of VPN Solutions Consultado en <https://openwrt.org/docs/guide-user/services/vpn/openvpn/performance>.
- [17] OpenVPN. Consultado en <https://en.wikipedia.org/wiki/OpenVPN>.
- [18] IPsec. Consultado en <https://www.rfc-editor.org/rfc/rfc6071>.
- [19] Using Advanced Encryption Standard (AES) CCM Mode with IPsec Encapsulating Security Payload (ESP). Consultado en <https://datatracker.ietf.org/doc/html/rfc4309>.
- [20] The Use of HMAC-SHA-1-96 within ESP and AH. Consultado en <https://datatracker.ietf.org/doc/html/rfc2404>.
- [21] IPsec en Routers AR Huawei, modo de encapsulación Consultado en <https://forum.huawei.com/enterprise/es/ipsec-en-routers-ar-huawei-modo-de-encapsulaci%C3%B3n/thread/539331-100235>.
- [22] Internet Key Exchange. Consultado en <https://www.rfc-editor.org/rfc/rfc2409>.
- [23] The Internet Key Exchange (IKE). Consultado en <https://datatracker.ietf.org/doc/html/rfc2409>.
- [24] 66 % of VPN's are not in fact broken. Consultado en <https://nohats.ca/wordpress/blog/2015/10/17/66-of-vpns-are-not-in-fact-broken/>.
- [25] Eyal Ronen, Adi Shamir Critical Review of Imperfect Forward Secrecy. Consultado en <https://www.wisdom.weizmann.ac.il/~eyalro/RonenShamirDhReview.pdf>.
- [26] Plex MKV Solution – A Complete Guide to Play MKV Videos in Plex Media Server. Consultado en <https://www.bluraycopys.com/resource/plex-play-mkv.html>.
- [27] Wake-on-LAN. Consultado en <https://www.codeproject.com/Articles/11469/Wake-On-LAN-WOL>.
- [28] Dynamic DNS (DDNS) on Debian Linux. Consultado en <https://www.linuxmaker.com/en/linux/dynamic-dns-ddns.html>.
- [29] Set up Dynamic DNS with IONOS. Consultado en <https://www.ionos.com/help/domains/configuring-your-ip-address/set-up-dynamic-dns-with-company-name/>.
- [30] Raspberry Pi Documentation. Consultado en <https://www.RaspberryPI.com/documentation/computers/os.html#voltages>.
- [31] GPIO: todo sobre las conexiones de la Raspberry Pi 4 y 3. Consultado en <https://www.hwlibre.com/gpio-raspberry-pi/>.
- [32] Setting up a Headless Raspberry Pi. Consultado en <https://www.RaspberryPI.com/documentation/computers/configuration.html#setting-up-a-headless-raspberry-pi>.
- [33] strongSwan_fail2ban. Consultado en https://github.com/s1nnerman89/strongSwan_fail2ban/blob/master/README.md.
- [34] COLISIONES EN MD5, SHA-1 y DES. Consultado en <https://1library.co/article/colisiones-md-sha-des-marco-te%C3%B3rico.q2690xez>.

- [35] Windows client cannot connect to StrongSwan: "EAP-Identity request configured, but not supported". Consultado en <https://superuser.com/questions/1348807/>.
- [36] About IPSec Algorithms and Protocols. Consultado en https://www.watchguard.com/help/docs/help-center/en-US/Content/en-US/Fireware/mvpn/general/ipsec_algorithms_protocols_c.html.
- [37] Configuring SSH Key Authentication on Linux. Consultado en <https://blog.knoldus.com/configuring-ssh-key-authentication-on-linux/>.
- [38] smb.conf Consultado en <https://www.samba.org/samba/docs/current/man-html/smb.conf.5.html>.
- [39] Installing on Debian and Ubuntu. Consultado en <https://www.webmin.com/deb.html>.
- [40] What is a GPG key, and how do I create it? Consultado en <https://www.quora.com/What-is-a-GPG-key-and-how-do-I-create-it>.
- [41] SSL Errors and HTTPS in Webmin. Consultado en <https://www.inmotionhosting.com/support/product-guides/cloud-server/ssl-errors-and-https-in-webmin/>.
- [42] How do I enable Wake On LAN? Consultado en <https://RaspberryPI.stackexchange.com/questions/126/how-do-i-enable-wake-on-lan>.
- [43] charon-cmd. Consultado en <https://docs.strongswan.org/docs/5.9/daemons/charon-cmd.html>.
- [44] About Diffie-Hellman Groups. Consultado en https://www.watchguard.com/help/docs/help-center/en-US/Content/en-US/Fireware/bovpn/manual/diffie_hellman_c.html.
- [45] IKEv2 Cipher Suites. Consultado en <https://docs.strongswan.org/docs/5.9/config/IKEv2CipherSuites.html>.
- [46] Así funciona el ataque TCP SYN, aprende cómo mitigarlo eficazmente. Consultado en <https://www.redeszone.net/tutoriales/seguridad/ataque-syn-que-es/>.
- [47] What's the difference between AES-CBC and AES-GCM. Consultado en <https://helpdesk.privateinternetaccess.com/kb/articles/what-s-the-difference-between-aes-cbc-and-aes-gcm>.
- [48] ChowEazyCopy. Consultado en <https://github.com/Cinchoo/ChoEazyCopy>.
- [49] Windows 10 IPSec VPN not respecting configured parameters (notably: encryption method). Consultado en <https://security.stackexchange.com/questions/239080/>.
- [50] Re: [strongSwan] ksoftirq thread reaching 100%. Consultado en <https://www.mail-archive.com/users@lists.strongswan.org/msg07386.html>.
- [51] Steffen Klassert. Parallelizing IPsec: switching SMP to 'On' is not even half the way. Consultado en https://www.strongswan.org/docs/Steffen_Klassert_Parallelizing_IPsec.pdf.

Glosario

- AES** Advanced Encryption Standard, tipo de cifrado establecido por el la organización estadounidense NIST. 16
- APU** Accelerated processing unit, nombre comercial a procesadores que combinan una unidad de procesamiento central con una unidad de procesamiento gráfico. 10
- ARM** Advance RISK Machines, arquitectura con un conjunto reducido de instrucciones para procesadores. 1
- CHAP** Challenge-Handshake Authentication Protocol, protocolo de autenticación utilizado originalmente por el protocolo punto a punto (PPP) para validar a los usuarios. Hoy en día también es utilizado por el protocolo PPTP. 16
- DDNS** Dynamic Domain Name Server, versión dinámica del DNS donde se actualizan los registros de dirección IP y dominio. 1
- DDoS** Distributed Denial-of-Service, ataque producido por varios sistemas que inundan el ancho de banda o los recursos de un sistema objetivo. 18
- DLNA** Digital Living Network Alliance, conjunto de reglas de interoperabilidad para compartir contenido multimedia entre dispositivos. 20
- DNS** Domain Name Server, sistema jerárquico descentralizado que traduce direcciones IP numéricas a nombres de dominios más fáciles para el ser humano. 1
- EAP** Extensible Authentication Protocol, conjunto de protocolos de autenticación utilizado con frecuencia en conexiones de red e Internet. 19
- ESP** Encapsulating Security Payload, cabecera del protocolo IPsec que añade autenticación y cifrado a los datos del paquete. 17
- IKE** Internet Key Exchange, protocolo para establecer una Security Association en IPsec. 18
- IPsec** Internet Protocol Security, protocolo seguro que autentica y cifra paquetes entre dos dispositivos a través de IP. 4
- L2TP** Layer 2 Tunneling Protocol, protocolo que permite implementar VPN pero que no cifra los paquetes de datos, solo los paquetes de control propios del protocolo. 16
- MKV** Matroska Multimedia Container, formato de archivo que puede contener una cantidad ilimitada de pistas de video, audio, imágenes o subtítulos en un archivo. Se trata de una especie de contenedor para video. 20

-
- NAT** Network address translation, método donde se mapea el espacio de dirección IP sobre otro modificando la dirección de red. 17
- NSA** National Security Agency, agencia de inteligencia del Departamento de Defensa de los Estados Unidos. 19
- PPTP** Point to Point Tunneling Protocol, protocolo obsoleto que permite implementar VPN. 16
- PSK** Pre-Shared Key, secreto compartido con anterioridad sobre un canal seguro. 16
- RSA** Rivest–Shamir–Adleman, sistema de cifrado con llave pública usa para la transmisión de datos de forma segura. 14
- SA** Security Association, establecimiento de elementos de seguridad compartidos entre dos dispositivos para soportar una comunicación segura. 18
- SMB** Service Message Block, protocolo de red a nivel de aplicación que permite el acceso compartido a archivos e impresoras entre dispositivos que pertenezcan a una misma red. 1
- SSH** Secure Shell Protocol, protocolo cifrado para operar sistemas informáticos de forma remota. 3
- TCP** Transmission Control Protocol, uno de los protocolos principales de IP que asegura la correcta recepción de los datos. 14
- WOL** Wake-on-LAN, estándar de ethernet que permite encender ordenadores mediante un mensaje de la red. 3

APÉNDICE A

Configuración del sistema

A.1 Dirección IP estática

Se modificará, o creará si no existe, el archivo «/etc/network/interfaces» y se añadirá lo siguiente.

```
auto eth0
iface eth0 inet static
address 192.168.0.226
netmask 255.255.255.0
gateway 192.168.0.1
dns-nameservers 1.1.1.1
dns-nameservers 1.0.0.1
```

A.2 Crea, eliminar y modificar usuarios

Siendo «pi» se ejecutará el comando y se rellenará la información:

```
# sudo adduser "nombre-del-usuario"
```

Después se añadirá este usuario al grupo «root» ejecutando:

```
# sudo usermod -a -G sudo "nombre-del-usuario"
```

A continuación se eliminará «pi» siendo el usuario anteriormente creado ejecutando:

```
# su "nombre-del-usuario"
# sudo userdel pi
```

Por último se cambiará la contraseña de «root»:

```
# sudo su
# passwd
```

A.3 Instalación y configuración de la VPN

Instalación de los paquetes necesarios para la VPN

```
# sudo apt update && sudo apt install strongswan strongswan-pki
```

Creación de certificado de autoridad público (ca-cert.pem) firmado por una llave privada (ca-key.pem).

```
# mkdir -p $HOME/pki/cacerts,certs,private
# chmod 700 /pki
# ipsec pki -gen -type rsa -size 4096 -outform pem >$HOME/pki/private/ca-k
# ipsec pki -self -ca -lifetime 3650 -in $HOME/pki/private/ca-key.pem
-type rsa -dn "CN=NOMBRE-QUE-QUERAMOS" -outform pem >$HOME/pki/cacerts/ca-
```

Creación del certificado de la VPN a partir de una llave privada (server-key.pem), el certificado de autoridad previamente hecho (ca-cert.pem) y la llave privada del certificado de autoridad (ca-key.pem). Es importante mencionar que los certificados van ligados al nombre del dominio, es decir, si hay más de un dominio que apunte a la misma dirección IP la VPN solo funcionará con el dominio que tenga en el certificado.

Ver el diagrama 6.6 para entender la relación entre certificados y llaves.

```
# ipsec pki -gen -type rsa -size 4096 -outform pem > /pki/private/server-k
# ipsec pki -pub -in $HOME/pki/private/server-key.pem -type
rsa | ipsec pki
-issue -lifetime 1825 -cacert $HOME/pki/cacerts/ca-cert.pem
-cakey $HOME/pki/private/ca-key.pem -dn "CN=NOMBRE-DOMINIO-DDNS"
-san "NOMBRE-DOMINIO-DDNS" -flag serverAuth -flag ikeIntermediate
-outform pem $HOME/pki/certs/server-cert.pem
# sudo cp -r $HOME/pki/* /etc/ipsec.d/
```

Configuración de la VPN a partir de un archivo de ejemplo del propio software.

```
# sudo mv /etc/ipsec.conf,.original
```

Añadir lo siguiente al archivo (Nota: el tutorial en el que me baso estaba mal en este apartado, ya que no ponía los algoritmos que puede aceptar el servidor y por tanto el servidor rechazaba cualquier conexión con un error de «NO CHOSEN SOL» y «NO PROPOSAL CHOSEN») por tanto esta configuración es propia para mi VPN.

```
# sudo nano /etc/ipsec.conf
config setup
charondebug='ike 1, knl 1, cfg 0'
uniqueids=no
conn ikev2-vpn
auto=add
compress=no
type=tunnel
```

```
keyexchange=ikev2
ike=aes256gcm16-sha384-sha256-sha-md5-modp2048-modp1024,aes256-sha384-sha2
fragmentation=yes
forceencaps=yes
rekey=no
left=%any
leftid=@NOMBRE-DOMINIO-DDNS
leftcert=server-cert.pem
leftsendcert=always
leftsubnet=0.0.0.0/0
right=%any
rightid=%any
rightauth=eap-mschapv2
rightsouceip=IP-VIRTUALES-QUE-QUERAMOS-OTORGAR
rightdns=IP-DEL-DNS-QUE-QUERAMOS
rightsendcert=never
eap_identity=%identity
```

Configuración de la autenticación de EAP para usuarios y una llave privada «RSA» para el propio servidor host.

```
# sudo nano /etc/ipsec.secrets
: RSA 'server-key.pem'
nombre-usuario : EAP 'contraseña'
nombre-usuario2 : EAP 'contraseña2'
nombre-usuario3 : EAP 'contraseña3'
# sudo systemctl restart strongswan
```

A.4 Instalación del paquete SAMBA

Este paquete software se instalará con el gestor de paquetes «APT».

```
# sudo apt install smb
```

Una vez instalado se debe configurar las particiones que se deseen. En la sección «global» se establecen reglas generales para todas las particiones, algunas de estas reglas son usuarios que tiene prohibido conectarse o el nivel de seguridad, en este caso a nivel de usuario.

En cambio las reglas de las otras particiones son solo para esa partición, las reglas más importantes son «create mask» y «directory mask», estas indican que permisos tendrán los archivos que se creen y que permisos tendrán los directorios que se creen respectivamente. En este caso en la partición [nas-user] los permisos para ambas máscaras son de «7-5-0», es decir total control para el autor («user»), lectura y ejecución para los usuarios pertenecientes al grupo «user» y nada para cualquier usuario ajeno.[38]

```

[global]
workgroup = CASA
security = user
invalid users = root bin daemon adm sync shutdown
guest account = nobody

[nas-user]
comment = carpeta compartida de user
path = /media/share/user
valid users = user
browsable = yes
read only = no
create mask = 0750
directory mask = 0750

[multimedia]
comment = carpeta de archivos multimedia (ESTA CARPETA ES
PÚBLICA)
path = /media/share/multimedia
browsable = yes
read only = no
create mask = 1777
directory mask = 1777

```

Por último la creación de los usuarios que usen SAMBA no solo se tiene que realizar en el propio sistemas UNIX, sino que también se deben añadir a la base de datos de SAMBA con el siguiente comando, si no se añadiese los usuarios no podrían iniciar sesión en SAMBA.

```
# sudo smbpasswd «nombre-usuario»
```

Después de ejecutar este comando se preguntará la contraseña que se quiere poner.

A.5 Instalación SSH y configuración

Para instalar y ejecutar el servidor SSH se hará ejecutando los siguientes comandos:

```
sudo apt install openssh sudo systemctl enable sshd sudo systemctl
start sshd
```

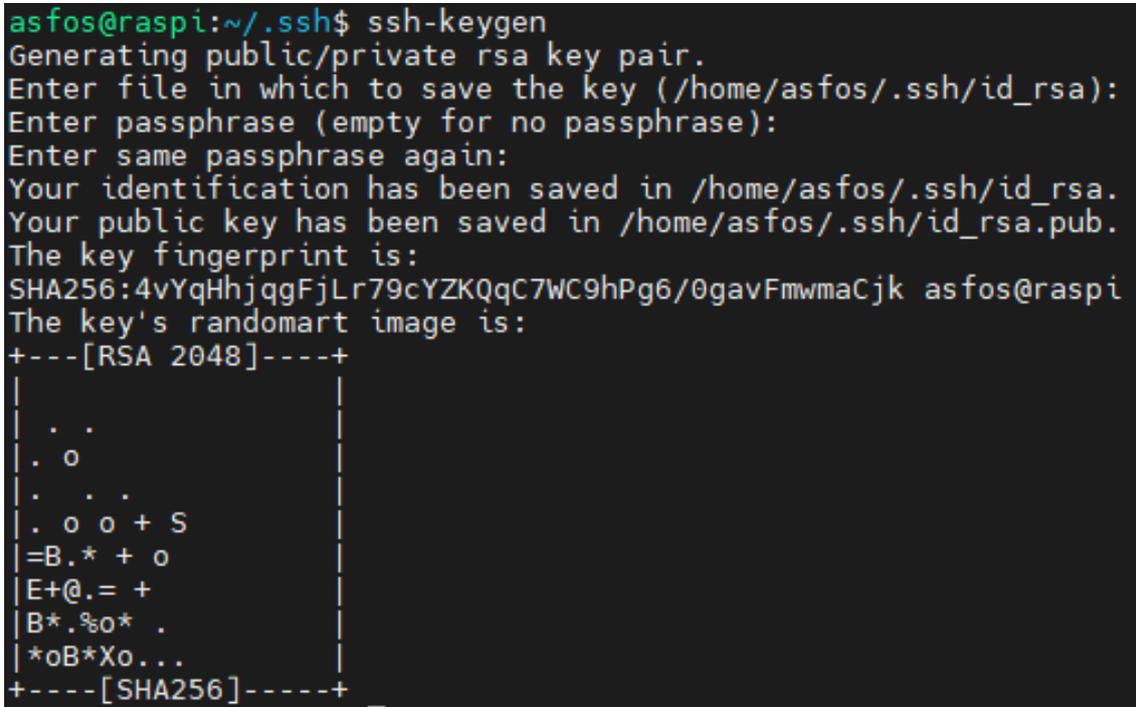
Una vez instalado se crearán los pares de llaves en los clientes mediante el comando `ssh-keygen` tal y como se muestra en la figura A.1 .

Una vez creadas las llaves configuraremos el servidor «OpenSSH». En este caso se ha optado por un método que solo permita el acceso a usuarios que pertenezcan a un grupo denominado «admin».

```

Include /etc/ssh/sshd_config.d/.conf
UsePAM yes
IgnoreUserKnownHosts no
StrictModes yes
PubkeyAuthentication no
PermitRootLogin no
PermitEmptyPasswords no
PasswordAuthentication no
GatewayPorts no
AllowTcpForwarding no
LoginGraceTime 120
KeepAlive yes
Protocol 1,2
Match Group admin
    PasswordAuthentication yes
    PubkeyAuthentication yes
    PermitEmptyPasswords no
    GatewayPorts no
    AllowTcpForwarding yes

```



```

asfos@raspi:~/ssh$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/asfos/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/asfos/.ssh/id_rsa.
Your public key has been saved in /home/asfos/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:4vYqHhjggFjLr79cYZKQqC7WC9hPg6/0gavFmwmaCjk asfos@raspi
The key's randomart image is:
+---[RSA 2048]-----+
|
|. .
|. o
|. . .
|. o o + S
|=B.* + o
|E+@.= +
|B*.%o* .
|*oB*Xo...
+----[SHA256]-----+

```

Figura A.1: Huella gráfica del par de llaves RSA de ejemplo

Para transferir entre clientes Linux simplemente se ejecutará el siguiente comando y se pondrá la contraseña del usuario con el que se quiere acceder al servidor.

```
sudo ssh-copy-id user@server
```

Para clientes Windows se deberá crear las llaves con otro software o incluso se puede desde páginas web «online». Estas llaves se copiarán en el directorio de Windows «C:/Users/user/.ssh» y posteriormente se copiará a mano, o mediante «scp» con contraseña, la llave pública (id_rsa.pub) al directorio .ssh del usuario, en este caso «/home/user/.ssh/id_rsa.pub».

Por último configuraremos que el usuario «root» no pueda ser accedido por SSH, esto se hará poniendo la variable «PermitRootLogin no» en el archivo «/etc/ssh/sshd_config»

A.6 Scripts

Para el «sniffer», se ha utilizado el programa «at» para poder planificar tareas, en este caso la tarea de suspender el ordenador, esto es necesario porque se requiere cierta histéresis entre conexiones y desconexiones, es decir, antes de apagar el ordenador esperar cierto tiempo por si se conecta otro usuario, si otro usuario llegara a conectarse esta suspensión se cancela. Por otra parte estas tareas de suspensión o cancelación se envían al NAS mediante SSH con intercambio de claves y un usuario con permisos específicos para este comando, no «root», tal y como se ve en el siguiente ejemplo.

Tarea de suspensión que se ejecutará dentro de 10 minutos.

```
ssh -i /path/a/las/llave/ssh/privada user@server.net "echo
'systemctl suspend' | at now +10 minutes"
```

Tarea de cancelación de suspensión.

```
ssh -i /path/a/las/llave/ssh/privada user@server.net "atrm
$(atq | awk 'print $1')"
```

Para la página web es más sencillo ya que el encendido y apagado del servidor son directos, sin esperas. Mediante una ejecución de «php» corremos un programa compilado en «C» que usa un «script» en «Bash» para encender o apagar el servidor que se indique por parámetro. El archivo compilado en C se ha hecho por temas de seguridad ya que para ejecutar los comandos del programa «wakeonlan» es necesario ejecutarlos como «root». De esta manera al ejecutar el «script» desde un archivo compilado evitamos la escalada de permisos o que se pueda ejecutar otros comandos como usuario «root» a través de «php».

APÉNDICE B

Configuración de elementos de seguridad

B.1 Reglas IPtables

Para poder obtener las reglas de IPtables se usa el siguiente comando:

```
sudo iptables -L
```

Existen tres cadenas «INPUT», «OUTPUT» y «FORWARD», las reglas que se usarán se añadirán en la cadena de «INPUT», paquetes que se reciben. Se añadirán dos reglas de aceptar paquetes UDP en dos puertos distintos, una regla para aceptar paquetes TCP para el SSH y para aceptar paquete de la página web.

Aceptar paquetes UDP por los puertos 500 y 4500.

```
sudo iptables -A INPUT -p udp -dport 500 -j ACCEPT
sudo iptables -A INPUT -p udp -dport 4500 -j ACCEPT
```

Aceptar paquetes para el protocolo SSH, TCP por el puerto 22 y aceptar paquetes TCP que provengan del puerto 22. Tanto para que se conecten al sistema como que el sistema se pueda conectar a otros dispositivos (necesario para los scripts explicados en 6.5).

```
sudo iptables -A INPUT -p tcp -dport 22 -j ACCEPT
sudo iptables -A INPUT -p tcp -sport 22 -m state --state ESTABLISHED
-j ACCEPT
```

Aceptar paquetes para HTTP, TCP puerto 80.

```
sudo iptables -A INPUT -p tcp -dport 80 -j ACCEPT
sudo iptables -A INPUT -p tcp -dport 80 -m state --state ESTABLISHED
-j ACCEPT
```

Por último para rechazar el resto de paquetes se cambiará la política por defecto a «DROP».

```
sudo iptables -P INPUT DROP
```

APÉNDICE C

Objetivos para el Desarrollo Sostenibles (ODS)

Este trabajo se cumple en varios objetivos: «Salud y bienestar», «Producción y consumo responsables» y «Acción por el clima».

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No procede
ODS 1. Fin de la pobreza				X
ODS 2. Hambre cero				X
ODS 3. Salud y bienestar			X	
ODS 4. Educación de calidad				X
ODS 5. Igualdad de género				X
ODS 6. Agua limpia y saneamiento				X
ODS 7. Energía asequible y no contaminante				X
ODS 8. Trabajo decente y crecimiento económico				X
ODS 9. Industria, innovación e infraestructuras				X
ODS 10. Reducción de las desigualdades				X
ODS 11. Ciudades y comunidades sostenibles				X
ODS 12. Producción y consumo responsables	X			
ODS 13. Acción por el clima		X		
ODS 14. Vida submarina				X
ODS 15. Vida de ecosistemas terrestres				X
ODS 16. Paz, justicia e instituciones sólidas				X
ODS 17. Alianzas para lograr objetivos				X

La privacidad es un elemento clave en la salud de las personas, sin ella las enfermedades mentales como el estrés o la ansiedad aumentarían. Es por esto que este trabajo tiene como una de sus intenciones ofrecer una herramienta para poder estar protegido en redes públicas, es decir, dar privacidad al usuario de la VPN. Además ofrece la comodidad de poder acceder a contenido personal gracias al servidor NAS de forma sencilla, incluso se pueden compartir estos contenidos si se utiliza un mismo NAS para distintos usuarios.

Por otro lado el consumo responsable es cada vez más relevante no solo por el cambio climático sino también por el aumento del coste de vida. Por estas razones es muy importante poder ahorrar la máxima cantidad de energía que no se esté utilizado y no desaprovecharla. Este ahorro de energía se consigue en este trabajo gracias al diseño del sistema híbrido, donde un ordenador que consume muy poco enciende, solo en los momentos necesarios, al ordenador más potente. Gracias a esto el gasto del ordenador potente no es en vano y solo consume cuando realmente se necesita. Este consumo res-

ponsable también repercute de manera indirecta en el bienestar de las personas ya que, teniendo en cuenta el precio actual de la electricidad, permite ahorrar dinero que de otra forma se habría malgastado.

Por último, el objetivo de «Acción por el clima» está íntimamente relacionado con el de «Producción y consumo responsable», ya que, si se hace un consumo responsable se gasta menos energía innecesariamente y por tanto se producen menos gases de efecto invernadero, reduciendo así el aumento del cambio climático. Este apartado es muy importante porque si no se trabaja de forma activa para reducir, ya que no se pueden evitar a estas alturas, las consecuencias del cambio climático, dentro de unos años será demasiado tarde para dar marcha atrás.

El resto de los «ODS» no se pueden relacionar con este trabajo por la propia naturaleza de estos. Este es un proyecto centrado para poder realizarse de forma doméstica y por tanto sus objetivos se centran en las necesidades de las familias, como la necesidad de un consumo bajo o alternativas baratas al uso del almacenamiento en la nube

Como conclusión decir que los «ODS» son fundamentales para el futuro y si no nos damos prisa en cumplirlos será demasiado tarde. Se supone que para 2030 los gases de efecto invernadero deberían reducirse un 50 por ciento, y a menos de ocho años de esta meta no se ven grandes avances en este apartado. Estos objetivos no son solo de la población, también tienen que intervenir los gobiernos y las empresas privadas porque si no, no se cumplirán a tiempo y aquí no hay un bando ganador.