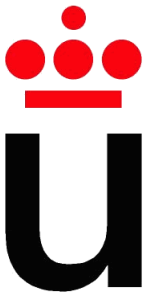

Universidad
Rey Juan Carlos

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

Curso académico 2004-2005

Proyecto Fin de Carrera

SERVICIO DE ACCESO REMOTO SEGURO PARA TERMINALES GSM Y GPRS

Autor: Jaime Pérez Crespo

Tutores: José Centeno González,

Miguel Ángel del Río Vega

Agradecimientos

Agradecer a José Centeno su esfuerzo y su constante predisposición, sin los que este proyecto no sería posible. A Antonio Tendero y Miguel Ángel del Río por la confianza y la ayuda depositada para este trabajo. A mis compañeros y amigos por sus críticas, sugerencias y ánimos, y por supuesto a mi familia.

Resumen

A lo largo de esta memoria se presenta una descripción del proceso de diseño e implementación de un servicio seguro de acceso remoto para aquellos usuarios de la Universidad Rey Juan Carlos que disponen de terminales con acceso a Internet mediante tecnologías *GSM* o *GPRS*. La idea fundamental de este proyecto, basado en una oferta realizada por Telefónica, consiste en proporcionar acceso a dichos usuarios a la red de la universidad, de forma que puedan utilizar sus servicios independientemente de su ubicación geográfica, y de forma totalmente fiable y segura tanto para ellos como para la propia universidad.

Para llevar a buen puerto este objetivo, se ha confiado en tecnologías para la creación y uso de Redes Privadas Virtuales, generalmente conocidas por las siglas *VPN*, basadas en criptografía y protocolos de autenticación del más alto nivel.

Inicialmente se ha realizado un estudio de las mencionadas tecnologías para evaluar la viabilidad y conveniencia de su uso en el caso que nos ocupa. Adicionalmente, y dado que el servicio ha sido implementado de forma conjunta con Telefónica, ha sido necesario cumplir con una serie de requisitos solicitados por dicha compañía, fundamentalmente relativos a la utilización de software homologado.

Una vez analizados los requisitos y funcionalidades impuestos para este proyecto, se ha procedido a la implementación del extremo del servicio correspondiente a la Universidad Rey Juan Carlos: un equipo terminador de túneles *IPSec* bajo el sistema operativo *GNU/Linux*, que además, y como se verá más adelante, realiza funciones adicionales derivadas de los cambios de requisitos por parte tanto de Telefónica como de la universidad.

Finalmente, se ha iniciado una fase de pruebas entre la propia universidad y Telefónica, a fin de corroborar el cumplimiento de las funcionalidades descritas y la estabilidad del servicio en general, para posteriormente situarlo en estado de producción.

Todo este trabajo ha derivado en un nuevo servicio que viene a completar los existentes y a facilitar el día a día de los usuarios con mayor movilidad de la universidad.

Índice general

Índice de figuras

Capítulo 1

Introducción

Las telecomunicaciones forman hoy día parte fundamental de la sociedad. La informática se ha ido arraigando en nuestros quehaceres cotidianos, hasta límites insospechados hace tan sólo diez años. En este contexto, es fácil imaginar la importancia de disponer de servicios informáticos y de telecomunicaciones adaptados a las necesidades personales y laborales de cada uno. En particular, y más aún en el caso que nos ocupa, se hace patente la necesidad de ofrecer servicios totalmente ubicuos en la medida de lo posible, esto es, usables sin importar nuestra localización ni otros detalles triviales.

Las corrientes informáticas actuales tienden a descentralizar los recursos en pos de un gran sistema, distribuído y omnipresente, que sea utilizable de cuantas formas pueda concebir el usuario. Esto derriba los conceptos clásicos de sistema informático y les infiere una nueva dimensión, valiéndose de los avances diarios en el campo de las redes y las telecomunicaciones. Más concretamente, Internet, la Red de redes, ha permitido desde hace unos años en adelante constituir un gigantesco sistema de servicios distribuídos a lo largo y ancho del planeta.

De forma aún más reciente, la explosión del mercado de las comunicaciones móviles y las distintas tecnologías que surgen entorno al mismo han hecho factible la posibilidad de comunicar dos puntos cualesquiera del globo casi inmediatamente. Satélites, radiofrecuencias... Múltiples formas de interconexión, cada una con dispositivos y escenarios de uso diferentes, permiten la utilización de sistemas informáticos en casi cualquier situación imaginable.

Gracias al abaratamiento de las conexiones a Internet y la magnitud de la interconexión conseguida, han surgido nuevos conceptos como el de *teletrabajo*. Se empieza a alcanzar el

objetivo de la deslocalización de recursos. Cada vez importa menos donde estés a la hora de utilizar el sistema. El usuario ya no tiene que lidiar con copias de los recursos para su propio uso, ni con un sistema monolítico que sólo pueda utilizar él. Ahora los usuarios no ven más que una interfaz del sistema, un terminal, un sencillo PC conectado a la red, que les permite trabajar utilizando los recursos de forma transparente.

Por supuesto, esta es la visión ideal del escenario particular que buscamos. En la práctica, resulta mucho más complejo de llevar a cabo por factores externos al propio avance tecnológico. La introducción de una red global de interconexión de ordenadores y otros dispositivos ha permitido derribar los obstáculos de la localización física, pero también ha introducido problemas nuevos en cuanto al acceso a la información se refiere. En esta sociedad la información es un bien muy preciado que mueve cantidades ingentes de dinero. Un bien cuya utilización está controlada fundamentalmente por motivaciones comerciales. Teniendo esto en cuenta, hay que proveer mecanismos de control de acceso a los recursos, la información, los servicios, que permitan identificar a los usuarios y mantener alejados a aquellos no autorizados.

La seguridad informática es el freno natural del avance exponencial del campo. De hecho, es precisamente la seguridad la responsable de que el objetivo que tratamos al inicio de este texto no se haya cumplido aún. Es necesario mantener un control absoluto sobre los recursos y sus usuarios a fin de evitar usos malintencionados o que puedan perjudicarnos de manera alguna. Y el primer paso para ejercer este control suele ser la localización de los usuarios y, por tanto, la imposibilidad de acceder al sistema desde donde deseemos.

Este es precisamente el problema que trataremos abordar y solventar en el presente proyecto. En el entorno de la Universidad Rey Juan Carlos, las necesidades de seguridad son muy altas, pero no deben cumplirse a costa de recortes en los servicios prestados. Por ello es necesario dar al personal de esta institución la posibilidad de trabajar desde cualquier ubicación requerida por las circunstancias. Continuaremos pues el trabajo [1] de Teófilo Romera en el Área de Comunicaciones de la Universidad para proporcionar un servicio de acceso remoto totalmente seguro mediante terminales *GSM* y *GPRS* a la red corporativa y sus servicios asociados.

[1] *Implantación de un servicio de acceso remoto seguro para la E.S.C.E.T.*, Proyecto de Fin de Carrera para la titulación de Ingeniería Informática de Teófilo Romera Otero, dirigido por José Centeno González.

En los siguientes capítulos se describirá en detalle el servicio prestado por *TME*[2], así como la definición, diseño e implementación de las infraestructuras necesarias por parte de la Universidad Rey Juan Carlos para la puesta en producción de dicho servicio.

Durante este capítulo introductorio trataremos dejar claros los conceptos básicos que se utilizarán a lo largo de esta memoria para explicar el proyecto. Formalizaremos definiciones de los protocolos y tecnologías utilizados, centrándonos en los puntos más destacados y relevantes que se han tenido en cuenta al diseñar el servicio.

A continuación, en un segundo capítulo, describiremos los objetivos iniciales y los cambios que han sufrido a lo largo de la ejecución del proyecto, así como todas las actividades que han implicado desde el inicio hasta la puesta en marcha del servicio.

En el capítulo tercero abordaremos la implementación realizada del servicio con todos los detalles necesarios, relativos a la configuración del equipo terminador de túneles, tanto a nivel de sistema operativo como de los servicios externos.

Por último, en el cuarto capítulo, analizaremos las conclusiones derivadas del desarrollo del proyecto y los posibles trabajos a realizar en un futuro partiendo de la base aquí obtenida.

1.1. Conceptos y definiciones

A continuación describiremos de forma más extensa las tecnologías clave en el desarrollo de este proyecto, ordenándolas por objetivos comunes. Pese a que la magnitud del proyecto es realmente grande en lo que a infraestructuras y distintas tecnologías se refiere, aquí nos centraremos fundamentalmente en aquello determinante para el desarrollo del mismo. Dejaremos un poco de lado otras partes fundamentales como pueden ser las tecnologías *GSM* y *GPRS*, que aunque son realmente clave, no han intervenido directamente de ningún modo en el presente estudio.

1.1.1. Redes privadas virtuales

El término *Red Privada Virtual*, frecuentemente referido por *VPN* [3], designa conceptos muy diferentes dependiendo del interlocutor que lo utilice. Por lo general, se asume un

[2] *Telefónica Móviles España.*
[3] Acrónimo del inglés *Virtual Private Network*

significado común y mínimo que lo designa como una extensión de una red local o privada a través de una red mayor y pública, como bien puede ser Internet. El objetivo básico, por tanto, consiste en proporcionar conectividad completa entre dos redes privadas diferentes y separadas, a través de una tercera, de modo que el usuario de cualquiera de estas redes realmente no pueda apreciar separación alguna. Paradójicamente, y como veremos más adelante, es posible utilizar redes privadas virtuales precisamente para todo lo contrario, separar una misma red en partes diferentes e independientes entre sí.

Por lo general cuando hablamos de redes privadas virtuales nos referimos a su implementación concreta más popular, unir redes locales a través de Internet de una forma segura. En los últimos años ésta se ha convertido sin duda en la aplicación más utilizada de este tipo de redes, debido a la rápida expansión y abaratamiento de las comunicaciones a través de una red global, *Internet*. Nótese que introducimos el concepto de seguridad del que no habíamos hablado en absoluto anteriormente, y que no está necesariamente ligado a las redes privadas virtuales.

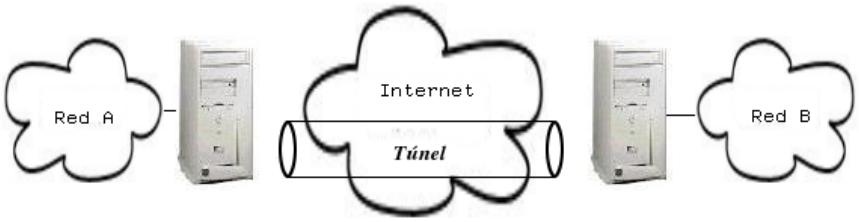

Figura 1.1: Esquema de una VPN a través de Internet. Se utiliza un medio de acceso público que permita la interconexión de las dos redes A y B de forma que se unan en una única red *virtual* cuyo acceso siga siendo restringido.

Podemos distinguir varios tipos de redes privadas virtuales atendiendo a la forma de implementarlas:

De acceso remoto: permiten a un usuario conectado a Internet desde algún punto remoto acceder a una red corporativa y sus servicios asociados. Un ejemplo de esto son trabajadores que realizan su actividad desde su propia casa, pero necesitan acceso a la red de su empresa. A través de su propia conexión a Internet establecen una red privada virtual que les permite trabajar como si estuviesen físicamente dentro de la red privada.

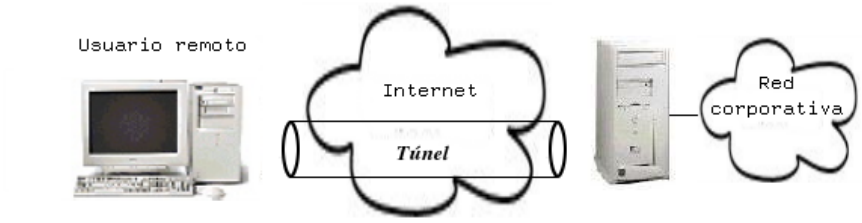

Figura 1.2: Esquema de una VPN de acceso remoto. Un usuario con conexión a Internet establece una VPN con la red corporativa, pudiendo acceder a la misma como usuario legítimo.

Punto a punto: unen distintas subredes de forma que la red privada resultante es el conjunto de todas ellas enlazadas en una VPN. Es un modelo típico de entidades bancarias, en el que las sucursales establecen una red privada virtual con un servidor central corporativo. Permiten eliminar los altos costes de infraestructuras en redes y vínculos punto a punto propios, dirigiendo todo el tráfico a través de una conexión a Internet mucho más barata.

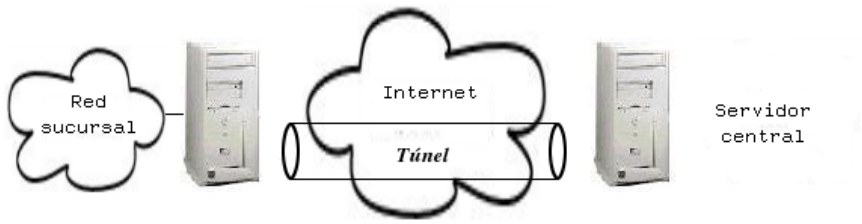

Figura 1.3: Esquema de una VPN punto a punto. La red local de una sucursal se conecta mediante una red privada virtual a un servidor central, que une respectivamente todas las sucursales en una misma red.

Internas: una aplicación realmente desconocida pero muy útil y potente consiste en establecer redes privadas virtuales dentro de una misma red local. El objetivo último es aislar partes de la red y sus servicios entre sí, aumentando la seguridad. Una aplicación muy típica de este modelo se utiliza para aumentar la seguridad en redes de acceso inalámbrico, separándolas así de la red física para evitar posibles fugas de información o accesos no autorizados.

Existen dos formas típicas de implementar una red privada virtual, basadas en *hardware* y en *software* respectivamente. En el fondo ambas soluciones implementan los mismos

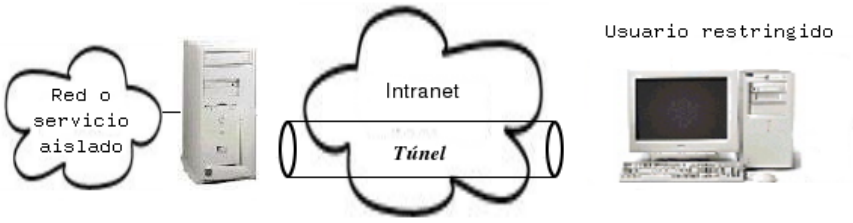

Figura 1.4: Esquema de una VPN interna. Se utiliza una VPN para restringir el acceso a ciertas partes de la red o usuarios concretos.

protocolos y soluciones típicas para construir redes privadas virtuales, aunque en distintos niveles. Lógicamente, una solución *hardware* ofrecerá un rendimiento mayor y permitirá construir sistemas complejos y grandes de forma sencilla. Como contrapartida, las soluciones *software* permitirán menor rendimiento, pero un aumento en la flexibilidad de configuración.

La mayoría de estos productos se centran en el actual estándar de facto para redes privadas virtuales, *IPSec*[4] , aunque cada vez aparecen más soluciones basadas en otros protocolos de distinto nivel, como *PPTP*[5], *L2TP* [6] o el cada vez más popular *SSL/TLS*[7]. Todos estos sistemas establecen *túneles* entre dos puntos de una red (los que son, estrictamente, los extremos de la red privada virtual) a través de los cuales se *encapsula* el tráfico intercambiado entre las dos redes a unir.

Estos últimos conceptos son la base teórica en la que se sustentan las redes privadas virtuales. El enlace se establece entre dos equipos que tienen acceso directo tanto a las subredes que se pretenden comunicar, como a la red que servirá de apoyo para la red privada virtual, como ya se ha dicho, generalmente Internet. Se establece una comunicación entre estos dos equipos, de forma que todo el tráfico que se dirige desde una de las subredes a la otra a través de la VPN es enviado desde el *terminador* de la primera al equipo *terminador* de la segunda.

Cuando un paquete cualquiera atraviesa la red privada virtual, el terminador saliente

[4] *IP Seguro*, en inglés, *IP Secure*.

[5] *Point-to-Point Tunneling Protocol*, *Protocolo de Tunelado Punto a Punto*.

[6] *Layer 2 Tunneling Protocol*, *Protocolo de Tunelado de la Capa 2*.

[7] *Secure Sockets Layer* (*Capa de Sockets Seguros*) es una tecnología ampliamente extendida y disponible en casi cualquier sistema operativo actual, por lo que cada vez existen más proyectos que se centran en esta solución al ser ideal para escenarios muy heterogéneos. OpenVPN es una de las implementaciones con más repercusión en la actualidad [5].

encapsula dicho paquete dentro de otro construido especialmente, correspondiente a alguno de los protocolos mencionados anteriormente, por norma general IPSec. El destino de ese paquete no será el destino original del paquete *encapsulado*, sino el terminador de la subred de destino. Una vez el paquete llega a través de la red pública al otro extremo de la VPN, el terminador correspondiente recoge dicho paquete y extrae su contenido intacto, redirigiéndolo tal cual a la red local de su extremo. Este es el concepto de *túnel* anteriormente mencionado, tráfico que se encapsula como carga útil de un protocolo utilizado para establecer la red privada virtual entre los dos terminadores del túnel, extremos de la VPN.

Figura 1.5: Concepto de tunelado. El paquete original es encapsulado dentro de un nuevo paquete en el terminador que lo envía a través de la VPN, y el otro terminador lo extrae, enviándolo tal cual a su destino.

Pese a que como se ha dicho, la funcionalidad principal que se espera de una red privada virtual es dar acceso a una red local desde puntos separados en el espacio, sin necesidad de realizar un gran desembolso en infraestructuras, se suele considerar también básico que la solución alternativa proporcione varios niveles de seguridad. En este sentido entendemos por seguridad dos aspectos fundamentales, la privacidad e integridad de las comunicaciones, y la autenticación de los usuarios y equipos que utilizan la red.

Dada la alta frecuencia con la que se utiliza Internet como medio sobre el que sustentar una VPN, es necesario establecer mecanismos adicionales que aseguren que el tráfico generado se mantiene fuera de los ojos de usuarios no autorizados. Al utilizar una red pública es imposible controlar qué uso se hace de las infraestructuras y quién tiene acceso a ellas. De hecho, ni siquiera es posible controlar cuáles son las infraestructuras que se utilizarán, que pueden variar a lo largo del tiempo. En estas condiciones es necesario recurrir a sofisticados esquemas criptográficos que aseguren la privacidad de las comunicaciones.

Más aún, resulta de interés garantizar que el tráfico no resulta alterado de su paso por la red pública de comunicaciones. Al igual que alguien podría monitorizar la red y obtener

información sensible correspondiente al tráfico generado por la VPN, el que ese tráfico viaje cifrado no asegura que esa misma persona no pudiese modificarlo de alguna forma. De modo que no sólo nos interesa cifrar los datos transmitidos, sino también buscar algún mecanismo que nos asegure que llegan a su destino tal cual salieron del origen.

Por otro lado, se hace patente la necesidad de identificar a los usuarios de la red y comprobar que su identidad sea correcta y adecuada para acceder a los recursos proporcionados. La aproximación típica que se utiliza para solventar este problema trabaja a dos niveles, autenticando por un lado el equipo informático desde el que se utiliza el servicio, y por otro lado al usuario que se encuentra detrás del mismo. Si alguno de estos dos elementos falla no podremos garantizar que alguien no autorizado acceda a los recursos, y por tanto la privacidad e integridad de las comunicaciones subyacentes no tienen sentido.

1.1.2. Confidencialidad e integridad

Como ya se ha mencionado, resulta básico hoy día el integrar por defecto la seguridad en las soluciones que implementan redes privadas virtuales a dos niveles. En el nivel de la confidencialidad e integridad de los datos que circulan por la red, la solución dominante hasta la fecha es IPSec.

IPSec

IPv4, el protocolo de red utilizado en Internet, carece por completo de cualquier tipo de seguridad en las comunicaciones. IPv6, la que está llamada a ser la siguiente versión del protocolo, sí proporciona en cambio un conjunto de medidas y protocolos para garantizar comunicaciones seguras en esta capa. Ese conjunto de medidas y protocolos es comúnmente conocido por *IPSec*, y se pueden implementar sobre IPv4 realizando apenas algunas modificaciones en la pila de TCP/IP del sistema operativo anfitrión.

IPSec proporciona autenticación, confidencialidad e integridad mediante diversos mecanismos. Fundamentalmente sus bases son tres protocolos diferentes:

AH: *Authentication Header* o *Cabecera de Autenticación*. Proporciona autenticación e integridad calculando un resumen sobre los paquetes a transmitir, pero no confidencialidad.

ESP: *Encapsulating Security Payload* o *Carga de Seguridad Encapsulada*. Proporciona autenticación, integridad y confidencialidad, utilizando mecanismos ágiles de cifrado cuya fiabilidad reside en el intercambio previo de claves.

IKE: *Internet Key Exchange* o *Intercambio de Claves en Internet*. Se trata de un protocolo para el intercambio de claves basado en el conjunto de reglas ISAKMP[8], y *Oakley*, una evolución del algoritmo *Diffie-Hellman* [9].

El funcionamiento de IPSec se basa en el intercambio de claves previo entre los extremos de una comunicación para el establecimiento de una *Asociación de Seguridad* o *SA*[10]. Esta necesidad de compartir información relativa a los interlocutores establece dos de las características claves de IPSec. Se trata de un protocolo unidireccional y orientado a conexión. Cada asociación de seguridad garantiza la seguridad en una dirección, por lo que si ambos interlocutores desean compartir información han de establecer cada uno su asociación de seguridad. Dicha asociación de seguridad se verifica en el campo SPI[11] que se incluye en las cabeceras IPSec, y que será comparado en el destino con su SAD[12], en la que se almacenan las asociaciones de seguridad conocidas en cada instante.

Una vez establecida dicha asociación, es posible enviar datos al otro extremo de dos formas diferentes. Bien modificando los paquetes IP originales para introducir las cabeceras adicionales, lo que se conoce como *modo transporte*, bien encapsulando el paquete original completo más las cabeceras AH o ESP dentro de uno nuevo, también denominado *modo túnel*.

En modo transporte, las cabeceras AH o ESP se insertan entre la propia cabecera IP y la carga útil del paquete (si se utiliza ESP, se incluyen cabeceras adicionales tras la carga útil del paquete).

En modo túnel se añaden los campos que correspondan según se utilice AH o ESP al paquete IP original, y dicho paquete se convierte en la carga útil de un nuevo paquete

[8] *Asociación de Seguridad de Internet y Protocolo de Gestión de Claves, Internet Security Association and Key Management Protocol.*

[9] Diffie-Hellman es un protocolo de intercambio de claves ampliamente extendido, creado en 1976, también conocido como *Intercambio Exponencial de Claves*. Existe numerosa información en la red sobre las bases matemáticas y las posibles implementaciones del algoritmo, por ejemplo, en la página web de RSA: `http://www.rsasecurity.com/rsalabs/`.

[10] *Security Association.*

[11] *Security Parameters Index, Índice de Parámetros de Seguridad*

[12] *Security Association Database, base de Datos de Asociaciones de Seguridad.*

IP con direcciones de origen y destino no necesariamente iguales a las originales. Este modo suele utilizarse cuando se dispone de al menos una pasarela de seguridad en uno de los extremos. Los paquetes viajan por la red sin que ningún dispositivo intermedio pueda acceder a la cabecera IP o los contenidos originales. Estas características hacen del modo túnel el ideal cuando se quiere crear una VPN sin que necesariamente ambos extremos de la comunicación tengan soporte específico para IPSec, ya que el túnel se puede crear en las mencionadas pasarelas de seguridad.

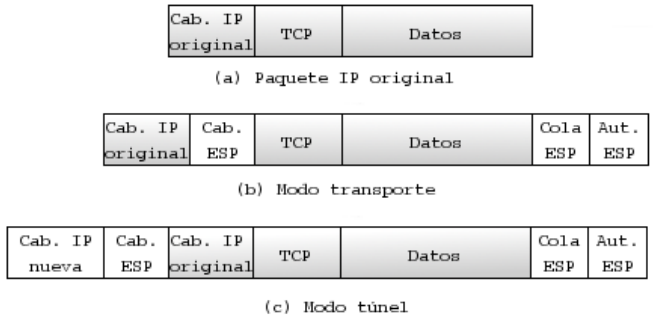

Figura 1.6: Modos de IPSec. En el modo transporte (b) el paquete original (a) es modificado añadiéndole campos AH o ESP. En el modo túnel (c) el paquete original (a) se encapsula dentro de un nuevo paquete IP con los campos AH o ESP.

El protocolo AH

La Cabecera de Autenticación o *AH*[13] proporciona en el ámbito de IPSec la autenticación del emisor y la integridad del mensaje mediante el cálculo de un código HMAC[14]. Una vez ambos extremos han establecido una SA, utilizan la clave acordada durante el intercambio inicial como clave simétrica con la que generar los resúmenes que se incluyen en las cabeceras. Sólo ambos extremos conocedores de la clave podrán calcular los resúmenes y verificar la integridad del paquete. Del mismo modo, un resumen que se corresponde con los contenidos de un paquete garantiza la autenticación del emisor, ya que sólo éste conocerá la clave con la que generar el resumen.

[13] *Authentication Header.*

[14] *Hash Message Authentication Codes*, o *Códigos de Autenticación del Mensaje por Dispersión.* El código HMAC se sirve de algoritmos como *MD5* o *SHA* (*Secure Hash Algorithm*, o *Algoritmo de Dispersión Seguro*) para obtener un *resumen* de la carga útil y algunas cabeceras sobre los que se aplica el algoritmo.

En AH, el resumen es calculado sobre la carga útil del paquete y las cabeceras estáticas del mismo, esto es, aquellas que no se modificarán durante el proceso, lo cual incluye las direcciones IP de origen y destino. Esto provoca que AH tenga graves problemas para tratar con NAT[15], ya que las pasarelas que utilizan este protocolo modifican la dirección IP de origen de los paquetes salientes, y la dirección IP de destino de los paquetes entrantes, acción que precisamente AH se encarga de detectar[16].

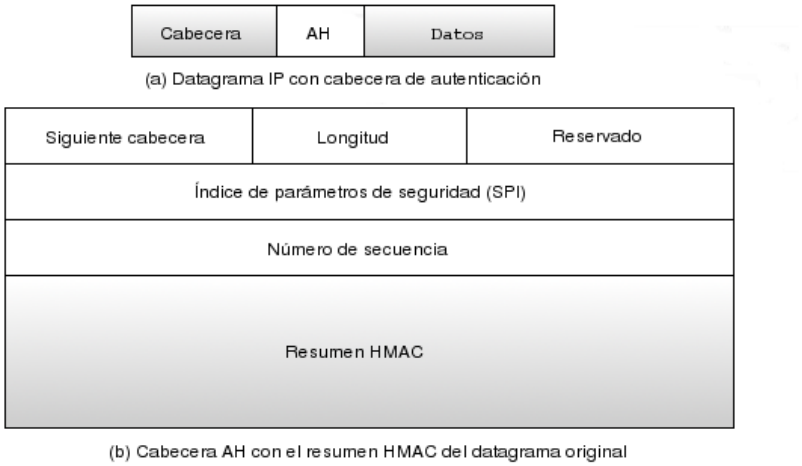

Figura 1.7: Cabeceras AH. Se insertan las cabeceras entre la cabecera IP original y la carga útil. Se calcula un resumen de las partes inmutables del paquete original que se incluye en la cabecera AH.

No entraremos a explicar en detalle las cabeceras utilizadas por este protocolo, mostradas en la figura 1.7. Puede ampliarse información en [9].

El protocolo ESP

El protocolo de Carga de Seguridad Encapsulada o *ESP*[17] proporciona, además de la integridad y autenticación de los datos transportados, la confidencialidad de los mismos. ESP se basa, al igual que AH, en el cálculo del código HMAC sobre el paquete, y adicionalmente en el cifrado del mismo. Dicho cifrado se realiza mediante algoritmos de clave simétrica, con la clave negociada para la asociación de seguridad vigente. El estándar de

[15] *Network Address Translation*, *Traducción de Direcciones de Red*. Es posible consultar el funcionamiento completo de este protocolo en [7].

[16] Existen muchos trabajos que tratan de lidiar con este problema, incluso una RFC que describe la compatiblidad necesaria entre IPSec y NAT [12].

[17] *Encapsulating Security Payload*.

IPSec obliga a la implementación de los algoritmos NULL[18] y DES [19]. En la actualidad, la mayoría de implementaciones de IPSec incluyen algoritmos mucho más potentes y actuales, como 3DES[20], AES [21] o Blowfish[22].

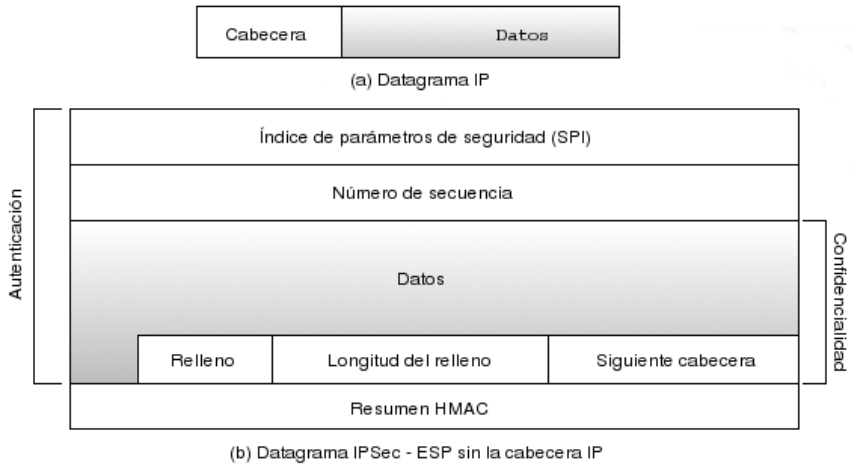

Figura 1.8: Cabeceras ESP. Los datos se rellenan hasta el tamaño de bloque. Los mismos datos y la cola ESP (el relleno, su longitud y el campo de siguiente cabecera) se autentican. Éstos, además de la cabecera ESP (el SPI y el número de secuencia) proporcionan autenticación. El resumen calculado del paquete se concatena con el mismo.

ESP encripta y calcula posteriormente el HMAC sobre el paquete IP original, exceptuando la cabecera IP. Esto hace que ESP no tenga los problemas antes mencionados de AH con NAT. Los datos se rellenan hasta completar una longitud múltiplo del tamaño de bloque utilizado. A continuación de ese relleno se incluyen dos campos, la longitud del

[18]NULL es un algoritmo que no hace nada. Existe una detallada descripción del mismo en la RFC 2410: `http://www.faqs.org/rfcs/rfc2410.html`.

[19]*Data Encryption Standard* es un algoritmo que utiliza claves de 56 bits para el cifrado. A día de hoy se le considera vulnerable a ataques de fuerza bruta. La descripción de este algoritmo en su implementación para ESP puede encontrarse en la RFC 1829: `http://www.faqs.org/rfcs/rfc1829.html`.

[20]Una mejora sobre el algoritmo ya comentado DES, consistente en aplicarlo en tres iteraciones. Su uso en DES se describe en la RFC 1851: `http://www.faqs.org/rfcs/rfc1851.html`.

[21]*Advanced Encryption Standard* es un avanzado algoritmo de criptografía simétrica con gran consideración en la actualidad, tanto que se ha convertido en el nuevo estándar en Estados Unidos. Se describe su implementación en IPSec en la RFC 3602: `http://www.faqs.org/rfcs/rfc3602.html`.

[22]Blowfish es un sistema criptográfico diseñado por Bruce Schneier para sustituir al obsoleto DES. Se basa en claves de longitud variable y bloques de 64 bis de tamaño para proporcionar una seguridad más que potente. Blowfish cuenta además con la ventaja de estar libre de ningún tipo de patente o licencia restrictiva, siendo así posible que el código fuente del algoritmo descrito por Schneier en `http://www.schneier.com/paper-blowfish-fse.html` esté disponible para descarga en su propia página web: `http://www.schneier.com/blowfish-download.html`.

mismo, y la siguiente cabecera dentro de la carga útil. Esta parte del paquete resultante se encripta, de forma que es imposible averiguar nada sobre los datos en sí mismos. El protocolo proporciona autenticación para esta parte del paquete y para el número de secuencia y el SPI en el resumen que se incluye al final del mismo.

Al igual que con AH, no entraremos a explicar detalladamente la función de las cabeceras que incluye ESP sobre un paquete IPSec, dejando al lector que amplíe la información necesaria en [10].

1.1.3. Autenticación y Autorización

Desde el punto de vista que nos ocupa, la autenticación se realiza en dos niveles separados, uno de ellos relativo al establecimiento de la asociación de seguridad de IPSec, y por otro lado a un nivel superior, el relativo a la autenticación del propio usuario del servicio. De ambos niveles se encargan los protocolos *IKE* y *Radius* respectivamente.

El protocolo IKE

El protocolo para Intercambio de Claves en Internet es el encargado en la infraestructura IPSec de proporcionar un entorno previo seguro para la compartición de una clave secreta y autenticación de los extremos. IKE utiliza el puerto 500 de UDP para establecer el intercambio de mensajes pertinente. Por lo general se implementa como una aplicación en espacio de usuario, al no formar parte del núcleo de muchos sistemas operativos.

Se compone de dos fases diferenciadas. La primera de ellas efectúa el intercambio de mensajes preeliminares necesarios, estableciendo una asociación de seguridad ISAKMP (*ISAKMP SA*). Este intercambio inicial puede estar basado en claves precompartidas o *PSK* [23], claves RSA para criptografía asimétrica, o una infraestructura PKI de certificados digitales[24].

Esta primera fase puede realizarse de dos formas diferentes, el modo normal y el modo agresivo. Ambos realizan las mismas tareas, aunque el modo agresivo emplea la mitad de

[23] *Pre Shared Keys*.

[24] *Public Key Infraestructure*. Este tipo de sistema se basa en la confianza en autoridades certificadoras, que emiten los llamados certificados, claves públicas firmadas digitalmente por estas entidades. De este modo, cuando se conoce la autenticidad de una autoridad certificadora y se puede verificar su firma digital (porque se dispone de su clave pública) se tiene la certeza de que la clave pública firmada es efectivamente de quien dice ser. Los estándares que describen esta infraestructura se pueden consultar en la RFC 3820: `http://www.faqs.org/rfcs/rfc3820.html`

mensajes que el modo principal para obtener los mismos resultados. La contrapartida es que este modo no proporciona autenticación de la identidad cuando se emplea junto con claves precompartidas (PSK), por lo que es vulnerable a ataques del tipo *hombre en el medio*[25]. Es por ello que generalmente se recomienda el uso de infraestructuras de clave pública si se va a utilizar el modo agresivo en una sesión IPSec.

En la segunda fase se utiliza la asociación de seguridad ISAKMP para establecer la asociación de seguridad IPSec definitiva, que determinará las claves a utilizar durante la sesión y otros parámetros de la misma. Por lo general se aprovechará esta fase para negociar dos asociaciones de seguridad, ya que, recordemos, en IPSec cada asociación es unidireccional.

Resulta extremadamente recomendable la lectura de la RFC 2409 [11] para la comprensión en detalle de las reglas ISAKMP, el algoritmo Oakley y en general el protocolo IKE, cuya implementación no es trivial.

Radius

Antes de hablar del Radius en sí mismo resulta conveniente introducir el marco *AAA* al lector, que implica una serie de conceptos básicos de los que se encarga el protocolo que nos ocupa. AAA son las siglas de *Authentication*, *Authorization* y *Accounting*, los tres aspectos de los que se ocupa la arquitectura:

Autenticación: es el proceso de verificar si la identidad de una persona o una máquina es efectivamente la que declara. Busca establecer una relación de confianza entre los interlocutores. Cuando hablamos de autenticar usuarios el primer ejemplo que se nos viene a la cabeza es el del nombre de usuario y la contraseña, aunque no todo es tan simple. Infraestructuras tan completas como los certificados digitales son soluciones más actuales y complejas al problema de la autenticación.

Autorización: involucra la utilización de reglas y plantillas para decidir si un usuario previamente autenticado goza de privilegios suficientes para acceder o no a un recurso. Por ejemplo, los permisos en un sistema de ficheros que determinan si un usuario puede leer, escribir o incluso ejecutar un archivo.

[25] *Man in the middle attacks*. El intruso se sitúa en un nodo intermedio de la comunicación, interceptando los mensajes, modificándolos si lo cree conveniente, y reenviándolos al otro extremo. De este modo los extremos no son conscientes de que se ha comprometido la comunicación.

Cuentas: entorno a la arquitectura AAA se encuentran las cuentas de usuario, que miden y documentan los recursos que un usuario utiliza durante su acceso. Por ejemplo, en un sistema UNIX es frecuente limitar a sus usuarios el número de procesos que pueden ejecutar concurrentemente, o la cantidad de CPU a utilizar.

Centrándonos en el aspecto que nos interesa de la arquitectura AAA, la autenticación se puede realizar siguiendo diversos esquemas:

Secuencia de agente: en esta secuencia, el servidor AAA actúa como delegado entre el equipamiento que presta el servicio[26] y el usuario final. El usuario contacta inicialmente con el servidor AAA, quien autoriza su petición y notifica al equipamiento de su decisión para que se le preste el servicio al usuario. El equipamiento del servicio notifica al servidor AAA cuando ha cumplido su petición, y el mismo servidor AAA notifica en última instancia al usuario.

Secuencia *de tiro* o *pull*: la más frecuente en servicios *dial* (de marcación telefónica) tradicionales. El usuario realiza la petición directamente al NAS y éste comprueba con el servidor AAA si debe proporcionar acceso.

Secuencia *de empuje* o *push*: esta secuencia, por contra, hace que el usuario pida algún tipo de certificación al servidor AAA, la cual deberá presentar más tarde al equipamiento que presta el servicio para garantizar su identidad y acceso al mismo.

La primera intuición hace pensar en usuarios, equipamientos y servidores de autenticación dentro de una misma red corporativa. El hecho es que esto no tiene por qué ser así en absoluto, y es muy común encontrar escenarios en los que existe la distinción entre empresa que contrata al cliente, y empresa que provee el acceso al servicio. Un ejemplo claro son los actuales proveedores de acceso a Internet en España, que suelen utilizar las infraestructuras propiedad de Telefónica para dar acceso al servicio a sus propios usuarios. El dueño de la infraestructura que llega al usuario tiene algún tipo de acuerdo con el prestador del servicio final de forma que el equipamiento del primero ha de ponerse en contacto con los servidores de autenticación del segundo. Este ejemplo ilustra muy por encima el concepto de *roaming*.

[26] Generalmente se denomina *NAS*, o *Network Authorization Server*, al equipamiento responsable de autenticar a un usuario en la red.

Radius es una implementación concreta de la arquitectura AAA, descrito en [8] con sumo detalle. Se encuentra ampliamente extendido al ser el primer protocolo de su tipo (de hecho, la especificación de Radius es previa al modelo AAA). Provee servicios de autenticación, autorización y cuentas de usuarios de forma genérica y completamente personalizable, y utiliza un esquema de autorización de secuencia *pull*, descrito ya previamente.

Pero como protocolo ya con bastantes años a sus espaldas tiene severas deficiencias que lo hacen cada día más reemplazable. Utiliza MD5 como algoritmo de dispersión para almacenar contraseñas, algoritmo que por otra parte se ha demostrado inseguro hace apenas unos meses. Tiene graves problemas de escalabilidad, admitidos ya en su propia RFC. Al estar basado en UDP y no implementar el concepto de *conexión*, no permite llevar ningún tipo de control sobre el uso de un servicio una vez el usuario ha sido autenticado.

Adicionalmente, Radius es un protocolo *salto a salto*[27], lo que quiere decir que cada servidor Radius en la cadena de autenticación tiene acceso a los datos de autenticación del usuario. Este modelo de seguridad puede parecer suficiente cuando se utilizan escenarios simples en los que no existe el concepto de *roaming*, pero la realidad es que por lo general las cadenas de autenticación son largas e implican diversos servidores de distintas empresas. En estas circunstancias, el modelo salto a salto es claramente inseguro.

Radius se sirve de identificadores conocidos como *realms*[28] para distinguir a los usuarios por tipos y saber en todo momento quién debe autenticarlos. Los realms pueden encontrarse en forma de sufijo o prefijo junto al nombre de usuario, separados por caracteres definibles, típicamente barras o arrobas. De esta forma un servidor Radius puede extraer patrones en los nombres de los usuarios en los que basarse a la hora de autenticar a un usuario. Por ejemplo, un servidor de la compañía Telefónica al que le llegue una petición de autenticación con un realm del tipo `usuario@telefonica.net` sabrá que para autenticar a ese usuario debe consultar, por ejemplo, una base de datos local. Por contra, si recibe un realm del tipo `usuario@urjc.es` sabrá inmediatamente que debe redireccionar la petición a uno de los servidores de autenticación de la Universidad Rey Juan Carlos que tenga configurados. Esta es la base conceptual que permite a Radius implementar esquemas complejos de roaming.

[27] *Hop to hop.*

[28] La traducción literal al castellano es "reino" "esfera". Una traducción más acorde con el sentido original inglés sería *dominio*, pese a que es propensa a confusión con el sistema de dominios de Internet.

Para información más detallada sobre este extenso protocolo sin duda conviene consultar su RFC [8]. [13] presenta además una excelente explicación de la arquitectura AAA en su conjunto y de una implementación concreta de Radius: FreeRadius.

Capítulo 2

Objetivos

2.1. Descripción del servicio

A lo largo de la introducción hemos hablado extensamente de las redes privadas virtuales y las nociones básicas que permiten implementarlas. Nos hemos centrado fundamentalmente en aquellos protocolos y arquitecturas que se han utilizado a lo largo de este proyecto, aunque sin olvidar las alternativas posibles.

Sin embargo no debemos asumir que el objetivo último consiste en la creación de una red privada virtual o VPN. Lo que aquí se presenta en su mayoría no son más que diversos mecanismos y tecnologías que nos permitirán ofrecer, de forma conjunta con Telefónica, el servicio de acceso remoto a la red de la universidad. Éste es, pues, el *leit motiv* de todo el trabajo realizado, ofrecer a nuestros usuarios la posibilidad de utilizar los recursos que la universidad pone a su disposición desde cualquier lugar y en cualesquiera circunstancias, sin dejar de lado, por supuesto, la seguridad de las comunicaciones. La movilidad absoluta de los empleados guiará por tanto todos nuestros esfuerzos.

2.1.1. Escenario de uso

La necesidad de construir una compleja infraestructura basada en redes privadas virtuales para permitir el acceso remoto a los teletrabajadores[1] puede resultar infundada a primera vista. Esto es así si nos dejamos guiar exclusivamente por las interpretaciones habituales de las VPN, en las que los servicios de acceso remoto suelen utilizarse para

[1]En textos en inglés es frecuente encontrarse el término *Road Warrior* para designar este mismo concepto.

proporcionar acceso desde el exterior a redes con direccionamiento privado. Este no es, obviamente, el caso de la universidad. A disposición de los usuarios hay múltiples rangos de direcciones IP de clase C que se utilizan casi en exclusiva. De este modo el direccionamiento en la inmensa mayoría de dispositivos utilizados en la universidad es público, y por contra son muy pocos los que utilizan direccionamiento privado, que se deja en exclusiva para cuestiones de administración de la propia red.

Los ordenadores que forman parte de la red de la universidad aprovechan este direccionamiento público para evitar, por un lado, los problemas que puede generar el uso de NAT, y por otro para garantizar que desde ellos se pueda ofrecer un servicio al exterior (Internet) cuando se requiera. Tal es el caso por ejemplo de los ordenadores de las *aulas Linux*, que permiten a los alumnos realizar sus prácticas de forma remota desde sus casas, sin saturar por ello un único servidor o interferir entre ellos. La única forma de conseguir esta flexibilidad es utilizar direccionamiento público siempre que sea posible, y no haya razones más poderosas que lo desaconsejen.

En estas circunstancias parece extraño utilizar una VPN para acceder de forma remota a la red, ya que, al fin y al cabo, el direccionamiento público ya permite ésto, siempre dentro de unos límites definidos por los administradores de cada servicio. La clave reside en que no todos los servicios que se prestan por o para la universidad permiten el libre acceso desde Internet. Algunos servicios no tienen razón de ser fuera de la red de la misma. Tal es el caso, por ejemplo, del CAU[2], que gestiona las incidencias y el soporte técnico general de todos los campus. Dado que el servicio es exclusivo para usuarios de la universidad, su uso está sujeto a un sistema de control de acceso que solo permite su utilización cuando el usuario se encuentra utilizando una dirección IP de la red interna. Normalmente esto sería equivalente a decir que el usuario se encuentre utilizando un equipo de la red de la universidad. Precisamente esta es la situación que cubre el presente trabajo, y que permite que los usuarios no tengan por qué utilizar sus equipos, ni siquiera estar presentes físicamente en la misma, para acceder a este tipo de servicios.

Adicionalmente, la universidad cuenta con servicios de terceros que se prestan a sus usuarios. El acceso a dichos servicios está también controlado mediante la dirección IP del usuario. Es el caso de suscripciones a revistras electrónicas u otros portales, como

[2] *Centro de Atención a Usuarios.*

Safari[3]. Lógicamente, es deseable que usuarios legítimos puedan beneficiarse de este tipo de servicios aunque no obtengan su acceso a Internet estando físicamente conectados a la red institucional.

2.1.2. Requisitos del servicio

Más allá del objetivo planteado para este trabajo es necesario establecer una serie de requisitos que determinen cómo encaminarlo. Dado que se trata de un proyecto conjunto con Telefónica, los requisitos vienen dados también por esta compañía, de forma que es necesario antenderlos antes de plantear requisitos propios.

Lamentablemente los requisitos suelen cambiar a lo largo de la ejecución de un proyecto, provocando que parte del trabajo se tenga que realizar de nuevo. Por supuesto esta no ha sido una excepción y tanto Telefónica como la universidad han realizado cambios en los requisitos iniciales que no estaban previstos. Muchas veces estos cambios vienen dados por decisiones técnicas que se descubren a lo largo del proyecto, si bien las decisiones puramente políticas también son habituales.

A continuación mostraremos los requisitos que han marcado el proyecto y cómo han variado durante el mismo, provocando que se ampliase su complejidad.

Ubicuidad

Tal como se ha comentado, el principal objetivo de este proyecto es conseguir que los usuarios de la red de la universidad Rey Juan Carlos puedan utilizar sus servicios sea cual sea su ubicación geográfica. Gracias a los terminales telefónicos GSM y GPRS, es posible conectarse a Internet desde cualquier lugar a través de ellos. Una vez establecida una conexión a Internet, es posible crear una red privada virtual entre la red de acceso de Telefónica y la propia red de la universidad, de forma que los usuarios puedan utilizarla transparentemente.

El usuario del servicio conectará su terminal a Internet siguiendo las instrucciones que le facilite el proveedor. Telefónica solicitará al usuario un nombre de usuario y una contraseña. Se le autenticará en la base de datos de la universidad, proporcionándole acceso

[3]Safari es un servicio que la editorial O'Reilly, junto con muchas otras, permite a sus usuarios la lectura y consulta sin restricciones de todas las publicaciones técnicas incluídas en su catálogo: `http://safari.oreilly.com/`.

o denegándolo. En caso afirmativo, la propia Telefónica le asignará una dirección IP de las destinadas por la universidad a tal efecto, y establecerá una VPN por la que el usuario tendrá acceso a los servicios internos. Con este sencillo procedimiento podemos eliminar las barreras geográficas, facilitando a los trabajadores su tarea, estén donde estén.

Facilidad de uso

Debemos atender al perfil del usuario típico de este servicio, que por lo general no tiene por qué gozar de grandes conocimientos informáticos. Más allá, es posible que sus conocimientos se limiten al mínimo que le permita realizar su trabajo, que no tiene por qué estar vinculado en absoluto a un entorno de nuevas tecnologías.

Estas circunstancias hacen deseable en gran medida que el servicio proporcionado sea lo más fácil de utilizar posible. Por ello merece la pena encaminar esfuerzos a que la infraestructura subyacente al servicio no modifique en modo alguno su uso, siendo así totalmente transparente al usuario. En las condiciones óptimas deseables, el usuario no debería notar diferencia alguna entre este servicio y cualquier otro servicio convencional de acceso a Internet mediante tecnología móvil.

Telefónica proporciona además distintas formas de autenticar a los usuarios. Algunas de ellas incluyen un identificador MSISDN[4], que se asocia con la tarjeta que utiliza el terminal telefónico. En nuestro escenario de uso estos terminales están sujetos a frecuentes cambios. Utilizar este tipo de restricciones obligaría al usuario del servicio a ser consciente de ellas cuando cambie su teléfono, algo que por supuesto no es deseable. Por eso se trata de mantener sencillez máximo, pidiendo al usuario tan sólo que recuerde un nombre y una clave de acceso.

Sencillez de implementación

Por supuesto, las cosas sencillas son generalmente las que más éxito tienen. Una implementación del servicio que no necesite recurrir a complicados procedimientos o técnicas hará posible reducir el número de problemas potenciales que pueden surgir. Cuanto más sencillo resulte implementar el servicio, más tiempo y esfuerzo se podrá dedicar a otros proyectos que puedan interesar a la universidad, redundando esto en el beneficio final de

[4] *Mobile Station Integrated Services Digital Network*. El número utilizado para designar a un suscriptor de telefonía móvil. Consiste en un código de país, un destino nacional, y un número de suscriptor.

los usuarios.

Facilidad de mantenimiento

En un entorno como el presente es habitual que exista un trasiego de empleados, becarios y otro personal que se tiene que adaptar rápidamente a los sistemas existentes. Merece la pena emplear el tiempo en facilitar lo máximo posible el mantenimiento del servicio y en su documentación. Esto no sólo ahorrará horas de trabajo al administrador en cada momento, sino que también facilitará a los recién llegados familiarizarse con el sistema y ponerse a trabajar lo antes posible.

Coste reducido

Los puntos anteriores suponen claramente una reducción notable de costes. El tiempo y esfuerzo que dedica un trabajador a una tarea se puede traducir directamente en dinero. Así mismo, las peculiaridades concretas de la implementación que se elija también influyen notablemente. No supone el mismo gasto económico la utilización de un PC convencional que utilice software libre para las tareas que debe realizar, que adquirir una solución propietaria dedicada, como por ejemplo los routers que existen en el mercado con funcionalidades para VPN. Por eso mantendremos en mente en todo momento la posibilidad de utilizar los recursos más económicos posibles sin pérdida de funcionalidad requerida.

Autenticación integrada

Una de las facilidades que da a elegir Telefónica a sus clientes a la hora de prestar el servicio es seleccionar el tipo de autenticación que desea. Es posible autenticar a los usuarios de dos formas diferentes:

- *Autenticación delegada*: la universidad facilita a Telefónica la relación de usuarios que utilizarán el servicio, indicando sus datos de identificación, y será ésta la que autentique a los usuarios, proporcionándoles o no acceso a la red privada. Este esquema tiene la ventaja de no requerir infraestructura de autenticación adicional por parte de la universidad, pero la desventaja de que los datos de autenticación dependan de terceros, y de no poder modificarlos cómo y cuándo se necesite.

- *Autenticación local*: en lugar de autenticar la propia Telefónica a los usuarios, escala las peticiones de autenticación a la universidad, siendo ésta la responsable en última instancia de decidir si un usuario accede o no al servicio. Supone un gasto mayor en infraestructuras al permitir a Telefónica consultas de autenticación sobre nuestros servidores, pero por contra facilita sobremanera la gestión de los usuarios.

Ni que decir tiene que para la universidad es mucho más ventajoso el modelo de autenticación local. La base de datos de usuarios ya existe [5] y también un sistema de autenticación y autorización de usuarios, mediante servidores Radius dedicados. En estas circunstancias, el coste de facilitar a Telefónica acceso a los servidores es mucho menor que el de mantener una relación de usuarios por cuenta ajena.

En este aspecto surgieron problemas a mitad de implementación. Telefónica no indicó explícitamente la necesidad de acceso directo a los servidores Radius, y esta posibilidad entraba en conflicto con la política de seguridad de la universidad. Posteriormente se decidió instalar un Radius que actuase como *proxy* o delegado para Telefónica, de forma que el acceso por parte de la misma a la autenticación fuese permanente y aislado, independientemente de los cambios que se puedan realizar en la infraestructura de autenticación global que se utiliza en la red interna. De este modo, se añade un nivel de seguridad adicional en las consultas Radius, y se asegura la prestación de las mismas a Telefónica ante eventuales cambios, por ejemplo un cambio en las direcciones IP de los servidores Radius corporativos.

Seguridad de acceso

De nada sirve proveer a trabajadores remotos de un acceso a la red de la universidad, fuertemente protegida, si ese acceso no es monitorizado y protegido también de la mejor manera posible. Por eso en todo momento se tiene en mente asegurar la confidencialidad de las comunicaciones, así como la autenticación de los usuarios del servicio.

La utilización de tecnologías específicas para VPN seguras como IPSec debe garantizar este punto.

[5] Conviene consultar el proyecto *Gente*: `http://gente.urjc.es`.

Equipación homologada

Telefónica, como prestador del servicio, requiere que la parte del mismo perteneciente a la universidad cumpla con ciertas propiedades acordes con sus sistemas.

- Terminador de túneles con IP pública y fija.
- Software Radius homologado:
  - Cisco Secure ACS 3.0.
  - Radiator Radius Server 3.0.
  - Shiva Access Manager 5.0p7.
  - ACE/Server v4.1.
  - Radius Server 2.1.
  - Lucent NavisRadius 4.1.3.
  - Steel-Belted Radius 3.0.
  - Radius de Telefónica Data 5.04.
  - Windows 2000 Server Radius.
  - Radius Access Policy Server.
  - GNU Radius 0.96.3.
  - Bay Secure Access Control Server 2.2.5.
- Terminador de túneles homologado:
  - Equipos Cisco.
  - Equipos Teldat.
  - Equipos Conectivity.
  - GNU/Linux.
  - VPN3000.
  - PIX.
  - Allied Telesyn.
  - W2000 / XP.

2.2. Estudio de alternativas

Atendiendo a los requisitos expuestos hasta el momento es necesario tomar decisiones para cada aspecto en el que se presentan diversas alternativas. A continuación exponemos las elecciones realizadas para las partes fundamentales del proyecto, divididas por categorías. Ha de tenerse en cuenta que todas estas decisiones parten de la base de la utilización de un PC convencional con GNU/Linux para actuar a modo de servidor, al ser la opción más barata, documentada y que ofrece mayor libertad de elección de las evaluadas.

2.2.1. Software para S/WAN

S/WAN o *Secure Wide Area Network* es el término que designa comúnmente redes de área ancha con seguridad incorporada, lo cual suele implicar el uso de IPSec. Como ya se ha dicho, IPSec es una extensión de IPv4 que garantiza seguridad a nivel de red. Sus implementaciones suelen requerir tanto cambios en la pila de protocolos TCP/IP del núcleo del sistema operativo, como utilidades que se ejecutan a un nivel mayor en espacio de usuario.

El proyecto por excelencia que implementa IPSec para sistemas Linux es *FreeS/WAN*[6]. Como característica a destacar sobresale la robustez de esta implementación y lo extendido de la misma durante años. FreeS/WAN es un proyecto muy grande que exige bastante dedicación. Es por ello que surgió algún proyecto para extender las funcionalidades de FreeS/WAN a una velocidad mayor que la habitual, ya que el desarrollo era demasiado lento en ocasiones. De este modo, mientras los desarrolladores de FreeS/WAN se dedicaban a mejorar el núcleo del proyecto, ganando en estabilidad y robustez, Super FreeS/WAN, agregaba cada vez más funcionalidades demandadas por los usuarios.

Esta situación se fue agravando paulatinamente por diferencias más bien políticas en el seno del grupo de desarrollo de FreeS/WAN, y por la aparición definitiva del núcleo 2.6 de Linux, que incluye soporte nativo para IPSec. El cúmulo de problemas derivó en el abandono del proyecto en el mes de abril del 2004 por parte de sus desarrolladores, liberando la que sería la última versión del mismo.

Dos proyectos surgieron a partir de FreeS/WAN continuando el camino abierto, openswan [2] y strongswan [4]. Ambos son un *fork* o bifurcación del proyecto original, con

[6] `http://www.freeswan.org/`.

objetivos diferentes para el futuro. Se mantienen muy actualizados (tanto el uno como el otro han liberado ya nuevas versiones en el año 2005) y llevan un buen ritmo de desarrollo. A día de hoy parece que openswan está ganando una mayor aceptación y número de usuarios, además de contar con el patrocinio de importantes empresas del sector como Novell o Suse.

En vista de la situación, y dado que ambos proyectos soportan a la perfección las funcionalidades que se les requieren para su cometido en este trabajo, la decisión de utilizar uno u otro estuvo meramente sujeta a la estabilidad del proyecto y los visos de continuidad de cada uno. Es por esto que se eligió finalmente openswan como implementación de IPSec, cuyo futuro parece garantizado gracias al patrocinio de Novell.

2.2.2. Software para encaminamiento

Telefónica utiliza el protocolo RIP[7] para mantener sus encaminadores dinámicamente, de forma que su configuración y actualización sea trivial. RIP es un protocolo de vectores de distancia, muy antiguo y desaconsejado actualmente, aunque no por ello deja de ser muy utilizado por su simplicidad frente a otros protocolos como OSPF[8] o BGP [9]. De hecho aún guarda una importante comunidad de defensores, y al fin y al cabo en sistemas muy complejos como Internet puede ser de gran ayuda.

Para dar soporte a este protocolo Telefónica recomienda la utilización del software de encaminamiento GNU Zebra[10]. El principal problema con este sistema es que, al igual que ocurre con FreeS/WAN, su desarrollo ha sido suspendido y recogido por diversos proyectos, por lo que no existe más soporte directo.

La alternativa elegida a Zebra es el software Quagga [3], una *suite* de software para encaminamiento que incluye soporte para los protocolos RIP, OSPF o BGP entre otras características, y que se basa en el antiguo GNU Zebra (de hecho, Quagga es un *fork* de éste). Nuevamente contamos con la ventaja de un desarrollo muy activo y al día, que establece la última versión de este software en el 31 de Enero del 2005.

[7] *Routing Information Protocol.*
[8] *Open Shortest Path First.*
[9] *Border Gateway Protocol.*
[10] `http://www.zebra.org`.

2.2.3. Software para Radius

La elección de un software Radius ha supuesto un severo problema durante el proyecto, al encontrar que los servidores FreeRadius que se utilizan en la red de la universidad no están homologados por Telefónica, lo cual ha obligado a cambiar de software instalando un servidor Radius proxy o delegado en el propio terminador de túneles. Lamentablemente, FreeRadius, que es el servidor más actualizado y completo actualmente, no es aconsejado por la operadora de telefonía, probablemente debido a su estado de *beta* casi permanente[11].

De los servidores Radius homologados y aconsejados, GNU Radius es de los más conocidos y utilizados, y mantiene los requisitos bastante aconsejables de coste reducido (es totalmente gratuito) y disponibilidad de las fuentes (como todo el software de GNU, está liberado bajo una licencia GPL). Pese al problema que supone el utilizar un servidor Radius diferente, y a la reducción de funcionalidades que provoca, supone una ventaja la estabilidad y solera de este proyecto, que lleva muchos años siendo ampliamente utilizado y probado.

2.2.4. Algoritmos

A lo largo de la introducción se han mencionado diversos algoritmos como parte de las implementaciones de IPSec y otras tecnologías, algoritmos fundamentalmente pensados para encriptar datos o calcular resúmenes de los mismos. IPSec permite una gran variedad en la elección de este tipo de algoritmos, y las implementaciones concretas suelen ser muy completas en este aspecto.

Aprovechando las pocas limitaciones que suelen existir en este aspecto, Telefónica proporciona a sus clientes la posibilidad de utilizar los algoritmos que prefieran. Esto incluye DES y 3DES para criptografía de clave simétrica, y MD5 o SHA para el cálculo de resúmenes de longitud fija.

En el caso de las funciones de *hashing* o dispersión, MD5 comienza a ser desaconsejado cuando interesa un nivel alto de seguridad. El 17 de Agosto del 2004, Xiaoyun Wang presentó un documento en el que demostraba que era posible el cálculo de colisiones en un margen muy razonable de tiempo. Esto echa por tierra las presunciones básicas de seguridad

[11] De hecho, hasta la fecha el proyecto FreeRadius todavía no ha liberado ninguna versión final de su software.

del algoritmo MD5, haciendo teóricamente posible encontrar distintos conjuntos de datos que originen un mismo resumen. Pese a esto, no está del todo claro hasta qué punto pueden afectar estos descubrimientos al uso de MD5 dentro de IPSec, y no existen casos reales de explotación de esta vulnerabilidad fuera de los laboratorios y la teoría. En cualquier caso, siempre resulta aconsejable evitar este tipo de problemas, ya que con toda seguridad investigaciones futuras que sigan esta línea derivarán en una demostración fehaciente de la inseguridad del algoritmo. Es por ello que se ha preferido la elección de SHA, que hasta la fecha es considerado totalmente fiable.

En la discusión entre DES y 3DES no existe duda posible. DES hace mucho tiempo que es considerado abiertamente inseguro y vulnerable a ataques de fuerza bruta, que cuando se dispone de recursos son tremendamente sencillos. Es por ello que resulta evidente la elección de 3DES, mucho más robusto y resistente a este tipo de ataques.

2.3. Metodología

Antes de entrar en detalles sobre la metodología de trabajo seguida en este proyecto, es necesario hacer notar que no se trata de un proyecto al estilo más tradicional. Este es un trabajo fundamentalmente de administración de sistemas y redes, y como tal, no ha involucrado directamente la creación de programas informáticos, si bien para ciertas tareas se han planeado pequeños scripts de automatización.

Partiendo de esta base, no es posible aplicar una metodología convencional como el *desarrollo en espiral* o *eXtreme Programming* a nuestro proyecto, sino que es necesario establecer una propia y personalizada que permita trabajar exactamente de la forma que necesitamos.

En concreto, se ha seguido un proceso iterativo de documentación, configuración y pruebas para cada una de las tecnologías que se ven involucradas en la ejecución de este proyecto. Cuando se quiere construir un sistema con tecnologías de vanguardia es necesario estar muy al día en lo que a novedades y alternativas se refiere. El mundo de las redes privadas virtuales e IPSec no es nuevo, pero si está en una continua renovación, lo que implica un estudio previo a cada implementación que se quiera realizar. Adicionalmente, como se ha visto, en este proyecto resulta fundamental la criptografía y algoritmos asociados frecuentemente a la misma. Este campo, extraordinariamente basado en las matemáticas,

sufre un vertiginoso y permanente cambio que provoca que algoritmos considerados seguros dejen de serlo en cuestión de horas.

Por esto el estudio y documentación previos para cada tipo de tecnología involucrada en el proyecto ha supuesto una parte clave del mismo. En este proceso de documentación se han utilizado diversos libros, fundamentalmente en lo que se refiere a aquellos protocolos y conceptos que más años tienen a sus espaldas, como son las VPN, IPSec o Radius. [13] en particular ha sido de una gran ayuda, ya que por desgracia este es un protocolo muy genérico que cuenta con una documentación muy pobre y sobre el que, por tanto, es extremadamente difícill obtener información precisa.

Resulta llamativo además cómo en la bibliografía adjunta predominan sobremanera páginas web y documentos que se pueden consultar por Internet sobre la documentación escrita. Esto es totalmente lógico si nos damos cuenta de que los libros dedicados a tecnologías tan punteras quedan rápidamente obsoletos.

Una vez obtenida una visión global de cada cosa, y la información necesaria para la comprensión de protocolos y servicios, se ha ido procediendo a configurar cada uno de ellos por separado. De este modo la tarea de montar un terminador de túneles IPSec se ha visto reducida a la solución de varios problemas de tamaño mucho menor, como el montaje y configuración del soporte IPSec, de un servidor proxy del protocolo Radius, o de un demonio de enrutado que soporte RIP. Esto facilita enormemente las cosas al poder enfocar todo el esfuerzo en puntos concretos a lo largo del tiempo.

Una vez se tiene el servicio configurado y en línea con todos los requisitos necesarios, se conciertan sesiones de pruebas con Telefónica para verificar el funcionamiento del terminador de túneles y del servicio en general. Terminado este periodo de exhaustivas pruebas, el servicio se situará en un estado de producción, pudiendo ser utilizado por fin por todo aquel que lo desee.

Por último, para el desarrollo de esta memoria y de la documentación que acompañará al sistema definitivo (recordemos que este proyecto se ha realizado sobre una maqueta de pruebas) se utiliza el sistema de creación de documentos LaTeX[12].

[12] `http://www.latex-project.org/`.

Capítulo 3

Descripción Informática

Una vez nos hemos familiarizado con los conceptos y tecnologías claves en el desarrollo de este proyecto, y hemos revisado brevemente cuáles han sido sus primeras fases de trabajo, vamos a centrarnos en los aspectos puramente técnicos de configuración del terminador de túneles IPSec.

Para la maqueta inicial que ha de servir para probar el servicio y habituarse a las herramientas que se utilizarán en el proyecto se ha utilizado un PC con la distribución Gentoo[1] Linux. Esta elección ha estado fundamentada en dos aspectos. Por un lado, gentoo es una distribución potente y flexible, que facilita sobremanera realizar la instalación de forma remota (lo cual resulta muy interesante al no contar el PC utilizado con monitor ni teclado), y que permite instalar un sistema mínimo totalmente personalizable por el administrador. Por otro lado, cuenta con un sistema de paquetes muy potente y actualizado, y numerosas herramientas para facilitar las tareas administrativas más comunes. De este modo, es posible centrarse exclusivamente en los aspectos de configuración de los servicios requeridos, sin tener que perder el tiempo con programas que no se van a necesitar o complicándose para realizar tareas rutinarias.

Más adelante, la máquina que provea definitivamente este servicio estará instalada bajo un sistema operativo Debian GNU/Linux, al ser éste el utilizado en la mayoría de servidores de la universidad, manteniendo así la homogeneidad y sencillez en el segmento de servicios.

[1] `http://www.gentoo.org/`.

3.1. Arquitectura

Dependiendo de la topología de la red y del rol que se quiera dar al equipo terminador de túneles IPSec dentro de la misma, es posible plantearse distintos tipos de arquitecturas. En particular, resultan evidentes dos implementaciones distintas.

3.1.1. Terminador puerta de enlace

Uno de los planteamientos posibles consiste en la construcción de un terminador de túneles IPSec que además sirva de puerta de enlace o *gateway* de la propia red local. Adicionalmente, podría actuar como cortafuegos entre la misma e Internet.

El PC utilizado tendría dos interfaces de red distintas. Una de ellas estaría conectada directamente al encaminador de salida a Internet, y sería la interfaz que soportaría la creación y mantenimiento de los túneles IPSec con los terminadores de Telefónica. La otra conectaría nuestro PC con la red local de la universidad, de forma que el tráfico se redireccionase lógicamente entre ambas interfaces.

Este diseño es conceptualmente muy sencillo, pero tiene el hándicap de provocar cambios colaterales muy agresivos. Implica cambiar la infraestructura general de la red y probablemente causar inconvenientes a los usuarios de otros servicios que nada tienen que ver con éste, durante el periodo de pruebas.

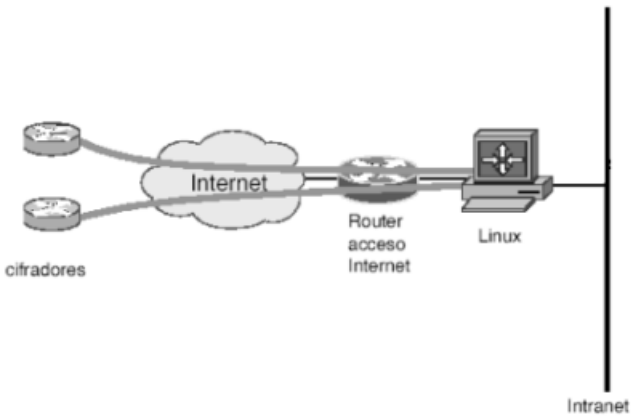

Figura 3.1: El terminador de túneles se sitúa como puerta de enlace de la red local, conectando una de las interfaces a Internet, y la otra a la propia red. De este modo el tráfico generado por los túneles IPSec se encamina al igual que cualquier otro, y se puede integrar fácilmente en las políticas de filtrado generales.

3.1.2. Terminador loopback

Como contrapartida, podemos diseñar una arquitectura en la que el PC terminador de túneles IPSec no sea más que otra máquina dentro de la red de servicios local. En este caso, el PC podría contar con tan sólo una interfaz de red conectada a la red local, por la que establecería los túneles IPSec y redireccionaría el tráfico al interior de la red, como si se tratase de una interfaz *loopback*.

De esta forma se aísla el servicio del resto, evitando afectar a todos los usuarios, y se facilita el montaje de la máquina, al no tener que realizar funciones extras a las mínimas requeridas. Es por ello que ésta es la arquitectura adoptada finalmente para el desarrollo del servicio, pese a que implica realizar un filtrado de los usuarios de la VPN por separado (en caso de que se desee dicho filtrado, por supuesto).

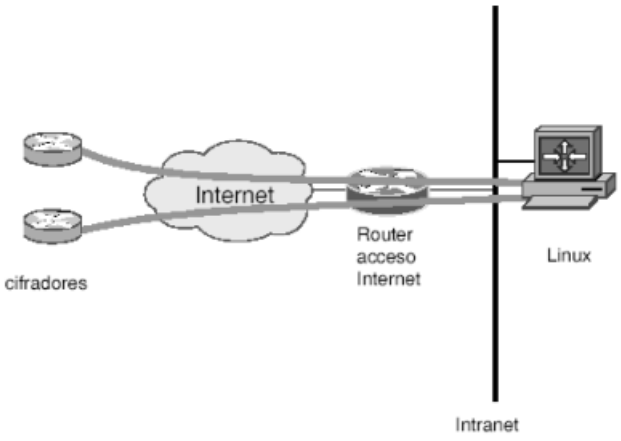

Figura 3.2: El terminador de túneles forma parte de la propia red. Los túneles se establecen contra su IP pública, y éste redirecciona el tráfico por su único interfaz a la red local, como si se tratase de un interfaz *loopback*.

3.2. Instalación y configuración del servidor

Para la instalación del sistema base que albergará los servicios necesarios es aconsejable seguir la documentación disponible en el excelente Gentoo Handbook [2]. Realmente merece la pena perder un poco de tiempo al principio en leer el manual, puesto que posteriormente nos será de gran ayuda ante posibles problemas.

[2] `http://www.gentoo.org/doc/es/handbook/index.xml`.

El PC utilizado es un Pentium II a 400MHz con 256MB de memoria RAM y aproximadamente 4GB de disco duro, dividido en cuatro particiones para `/boot`, `/`, `/usr` y `/var`, respectivamente.

3.2.1. El núcleo y el sistema base

Una vez instalada la base de paquetes mínimos que requiere un sistema linux para funcionar de forma adecuada, nos ocupamos de conseguir un núcleo con el soporte de IPSec para el correcto funcionamiento de openswan. Instalamos las fuentes del kernel 2.4.26 con los parches de seguridad de gentoo y los parches para IPSec ya incluidos de serie, y verificamos las siguientes opciones en su configuración, dentro del submenú `Networking options`:

```
<M> IP Security Protocol (FreeS/WAN IPSEC)
--- IPSec options (FreeS/WAN)
[*]    IPSEC: IP-in-IP encapsulation (tunnel mode)
[*]    IPSEC: Authentication Header
[ ]       HMAC-MD5 authentication algorithm
[*]       HMAC-SHA1 authentication algorithm
[*]    IPSEC: Encapsulating Security Payload
[*]       3DES encryption algorithm
[ ]    IPSEC: IP Compression
[ ]    IPSEC Debugging Option
[ ] IPSEC NAT-Traversal
```

Nótese que damos soporte al modo túnel que vamos a utilizar, y al protocolo ESP. Si no marcamos la opción 3DES, se utilizaría el protocolo por defecto, que es el ya comentado DES. Así mismo, es necesario marcar el soporte para AH ya que dentro de éste están incluidas las opciones del cálculo de HMAC. En particular, como ya hemos visto previamente, nos interesa utilizar el algoritmo SHA para cálculo de resúmenes. No nos interesan opciones como la compresión IP o el soporte para *NAT-Traversal*, ya que como hemos visto no realizaremos NAT en ninguno de los extremos del túnel.

Compilamos el núcleo y lo instalamos de la forma habitual, de modo que al arrancarlo tengamos el soporte para IPSec necesario. No entraremos en detalle sobre el resto de configuración del núcleo. Baste decir que procuraremos dar soporte al hardware directamente

en el núcleo, y añadiremos las opciones correspondientes a iptables[3] para su posterior uso.

Configuración de red

El equipo terminador de túneles IPSec cuenta con una sola interfaz de red física conectada a la red privada de la universidad, y con una sola dirección IP pública con la que accede tanto al exterior como al interior de la red. La configuración concreta es la siguiente:

```
minimum root # ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:00:24:C8:0A:98
          inet addr:193.147.184.193  Bcast:193.147.184.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:14905358 errors:3 dropped:0 overruns:0 frame:0
          TX packets:534587 errors:3999 dropped:0 overruns:0 carrier:3999
          collisions:32135 txqueuelen:1000
          RX bytes:1438569655 (1371.9 Mb)  TX bytes:48817647 (46.5 Mb)
          Interrupt:5 Base address:0xe800
minimum root # route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
193.147.184.0   0.0.0.0         255.255.255.0   U     0      0        0 eth0
127.0.0.0       127.0.0.1       255.0.0.0       UG    0      0        0 lo
0.0.0.0         193.147.184.1   0.0.0.0         UG    0      0        0 eth0
```

Para que esta configuración se aplique automáticamente al arrancar la máquina, modificamos el fichero de configuración `/etc/conf.d/net`:

```
# configuración del interfaz de red
iface_eth0="193.147.184.193 broadcast 193.147.184.255 netmask 255.255.255.0"
# gateway por defecto de la red
gateway="eth0/193.147.184.1"
```

Gentoo proporciona scripts para activar o desactivar los interfaces de red de forma automática. Utilizándolos resulta trivial conseguir que esta configuración se aplique en el arranque del PC:

[3]IPTables es el sistema de filtrado de paquetes y control de red por defecto en los núcleos 2.4 y posteriores de Linux. Sustituye al ya obsoleto *ipchains* de los núcleos 2.2 y se centra en la creación de reglas que decidan qué hacer con los paquetes a nivel de red y de transporte. Se puede encontrar más información en la página web del proyecto: `http://www.netfilter.org/`.

```
minimum root # rc-update add net.eth0 default
```

Una vez configurada la red, podemos proceder a configurar el cortafuegos que filtrará el tráfico del terminador de túneles. Para empezar, es fundamental activar la opción del núcleo de redirección de IP (*IP Forwarding*) y desactivar la comprobación de la ruta de origen de los paquetes IP (sin hacer esto último IPSec no funcionará correctamente). Podemos hacerlo interactivamente de forma sencilla:

```
minimum root # echo 1 > /proc/sys/net/ipv4/ip_forward
minimum root # for f in /proc/sys/net/ipv4/conf/*/rp_filter
> do echo 1 > $f
> done
```

Nuevamente, deseamos que estos cambios se realicen de forma automática al arrancar el servidor, sin necesidad de la intervención de un operador, por lo que modificamos el fichero `/etc/sysctl.conf`:

```
# Habilitamos la redireccion de paquetes
net.ipv4.ip_forward = 1
# Deshabilitamos ECN
net.ipv4.tcp_ecn = 0
# Habilitamos la verificacion de la ruta origen
net.ipv4.conf.default.rp_filter = 0
```

Nos aseguramos de que el módulo *ipsec* (recordemos que hemos habilitado el soporte IPSec en el núcleo en forma de módulo, no como parte del mismo. En caso contrario no sería necesario este paso) se cargue al arrancar el sistema operativo:

```
minimum root # echo ipsec >> /etc/modules.autoload.d/kernel-2.4
```

Dado que como ya se ha dicho, la maqueta no cuenta con teclado ni monitor conectados, la única forma de trabajar con ella se reduce a alguna forma de acceso remoto. En particular SSH[4] es fundamental para nuestros propósitos, por lo que instalamos un servidor:

[4] *Secure SHell*. Se trata de un protocolo de conexión segura entre dos máquinas de la red que utiliza criptografía avanzada y permite la ejecución remota de comandos.

```
minimum root # emerge openssh
```

y modificamos su fichero de configuración `/etc/ssh/sshd_config` adecuadamente:

```
Port 22
Protocol 2
ListenAddress 193.147.184.193

# Logging
#obsoletes QuietMode and FascistLogging
SyslogFacility AUTH
LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes

# Set this to 'yes' to enable PAM authentication (via challenge-response)
# and session processing. Depending on your PAM configuration, this may
# bypass the setting of 'PasswordAuthentication' and 'PermitEmptyPasswords'
UsePAM yes

# override default of no subsystems
#Subsystem      sftp    /usr/lib/misc/sftp-server
```

Lo arrancamos y le indicamos al sistema que lo arranque al iniciarse el sistema operativo:

```
minimum root # /etc/init.d/sshd start
 * Starting sshd...                                    [ ok ]
minimum root # rc-update add sshd default
```

En todo momento podemos comprobar el estado de los servicios del nivel por defecto de ejecución con el comando `rc-status`. Una vez llegados a este punto, configuraremos un cortafuegos mediante iptables para asegurar los accesos al PC terminador de túneles. Creamos un sencillo script que nos permita modificar rápidamente estas reglas, y lo probamos simplemente ejecutándolo:

```
#!/bin/bash
IPTABLES='/sbin/iptables'
# Interfaces
EXTIF='eth0'
IPSECIF='ipsec+'
# IP's de interés
PUBIP='193.147.184.193'
# Terminadores túneles IPSec Telefónica
TERM1='195.55.47.10'
TERM2='195.55.47.11'
# Radius de la universidad
RADIUS1='193.147.184.3'
RADIUS2='193.147.184.22'
# DNS de la universidad
DNS1='193.147.184.2'
DNS2='193.147.71.64'
# Pools de usuarios de la VPN
POOLGSM='193.147.59.160/27'

# Habilitamos IP forwarding en el núcleo
/bin/echo 1 > /proc/sys/net/ipv4/ip_forward

# Eliminar reglas y cadenas
$IPTABLES -F
$IPTABLES -X

##
## ADMINISTRACIÓN DE SERVICIOS
##
# Acceso para el servicio ssh
$IPTABLES -A INPUT -s 193.147.184.0/24 -d $PUBIP -i $EXTIF \
 -p tcp --sport 1024:65535 --dport 22 -j ACCEPT
$IPTABLES -A INPUT -s 193.147.72.0/24 -d $PUBIP -i $EXTIF \
 -p tcp --sport 1024:65535 --dport 22 -j ACCEPT

# Acceso a los interfaces telnet de administración de zebra y ripd
$IPTABLES -A INPUT -s 193.147.184.0/24 -d $PUBIP -i $EXTIF \
 -p tcp --sport 1024:65535 --dport 2601 -j ACCEPT
$IPTABLES -A INPUT -s 193.147.72.0/24 -d $PUBIP -i $EXTIF \
 -p tcp --sport 1024:65535 --dport 2601 -j ACCEPT
$IPTABLES -A INPUT -s 193.147.184.0/24 -d $PUBIP -i $EXTIF \
 -p tcp --sport 1024:65535 --dport 2602 -j ACCEPT
$IPTABLES -A INPUT -s 193.147.72.0/24 -d $PUBIP -i $EXTIF \
 -p tcp --sport 1024:65535 --dport 2602 -j ACCEPT
```

```
# Bloqueamos el acceso desde la VPN a los servicios del gateway
$IPTABLES -A INPUT -d $PUBIP -i $IPSECIF -p tcp \
 --dport 22 -j DROP
$IPTABLES -A INPUT -d $PUBIP -i $IPSECIF -p tcp \
 --dport 2601 -j DROP
$IPTABLES -A INPUT -d $PUBIP -i $IPSECIF -p tcp \
 --dport 2602 -j DROP

##
## Servicios públicos
##
# Acceso UDP al demonio RIP
$IPTABLES -A INPUT -d $PUBIP -i $EXTIF -p udp \
 --dport 520 -j ACCEPT

# Acceso consultas Radius
$IPTABLES -A INPUT -s $TERM1 -d $PUBIP -p udp \
 --dport 1812 -j ACCEPT
$IPTABLES -A INPUT -s $TERM2 -d $PUBIP -p udp \
 --dport 1812 -j ACCEPT
$IPTABLES -A INPUT -s $RADIUS1 -d $PUBIP -p udp \
 --dport 1812 -j ACCEPT
$IPTABLES -A INPUT -s $RADIUS2 -d $PUBIP -p udp \
 --dport 1812 -j ACCEPT

# Permitimos protocolo ESP de IPSec
$IPTABLES -A INPUT -s $TERM1 -d $PUBIP -p esp -j ACCEPT
$IPTABLES -A INPUT -s $TERM2 -d $PUBIP -p esp -j ACCEPT

# Permitimos las transacciones IKE
$IPTABLES -A INPUT -s $TERM1 -d $PUBIP -p udp --sport 500 \
 --dport 500 -j ACCEPT
$IPTABLES -A INPUT -s $TERM2 -d $PUBIP -p udp --sport 500 \
 --dport 500 -j ACCEPT

# Permitimos el tráfico entre las VPN y la red de la universidad
$IPTABLES -A INPUT -d $POOLGSM -i $EXTIF -j ACCEPT
$IPTABLES -A INPUT -s $POOLGSM -i $IPSECIF -j ACCEPT

# Permitimos consultas DNS con los servidores privados
$IPTABLES -A INPUT -s $DNS1 -d $PUBIP -i $EXTIF -p udp \
 --sport 53 -j ACCEPT
$IPTABLES -A INPUT -s $DNS2 -d $PUBIP -i $EXTIF -p udp \
 --sport 53 -j ACCEPT
```

```
# Permitimos el tráfico originado desde el terminador
# (conexiones TCP ya abiertas)
$IPTABLES -A INPUT -d $PUBIP -i $EXTIF -p tcp \
 ! --tcp-flags SYN,RST,ACK SYN -j ACCEPT

# Bloqueamos todo el tráfico
$IPTABLES -A INPUT -i $EXTIF -p tcp -j DROP
$IPTABLES -A INPUT -i $EXTIF -p udp -j DROP
```

Verificamos el correcto funcionamiento de la red una vez ejecutado este script, y comprobamos que las reglas se han aplicado correctamente:

```
minimum root # iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  193.147.184.0/24     193.147.184.193
     tcp spts:1024:65535 dpt:22
ACCEPT     tcp  --  193.147.72.0/24      193.147.184.193
     tcp spts:1024:65535 dpt:22
ACCEPT     tcp  --  193.147.184.0/24     193.147.184.193
     tcp spts:1024:65535 dpt:2601
ACCEPT     tcp  --  193.147.72.0/24      193.147.184.193
     tcp spts:1024:65535 dpt:2601
ACCEPT     tcp  --  193.147.184.0/24     193.147.184.193
     tcp spts:1024:65535 dpt:2602
ACCEPT     tcp  --  193.147.72.0/24      193.147.184.193
     tcp spts:1024:65535 dpt:2602
DROP       tcp  --  0.0.0.0/0            193.147.184.193
     tcp dpt:22
DROP       tcp  --  0.0.0.0/0            193.147.184.193
     tcp dpt:2601
DROP       tcp  --  0.0.0.0/0            193.147.184.193
     tcp dpt:2602
ACCEPT     udp  --  0.0.0.0/0            193.147.184.193
     udp dpt:520
ACCEPT     udp  --  195.55.47.10         193.147.184.193
     udp dpt:1812
ACCEPT     udp  --  195.55.47.11         193.147.184.193
     udp dpt:1812
ACCEPT     udp  --  193.147.184.3        193.147.184.193
     udp dpt:1812
ACCEPT     udp  --  193.147.184.22       193.147.184.193
     udp dpt:1812
ACCEPT     esp  --  195.55.47.10         193.147.184.193
```

```
ACCEPT     esp  --  195.55.47.11         193.147.184.193
ACCEPT     udp  --  195.55.47.10         193.147.184.193
     udp spt:500 dpt:500
ACCEPT     udp  --  195.55.47.11         193.147.184.193
     udp spt:500 dpt:500
ACCEPT     all  --  0.0.0.0/0            193.147.59.160/27
ACCEPT     all  --  193.147.59.160/27    0.0.0.0/0
ACCEPT     udp  --  193.147.184.2        193.147.184.193
     udp spt:53
ACCEPT     udp  --  193.147.71.64        193.147.184.193
     udp spt:53
ACCEPT     tcp  --  0.0.0.0/0            193.147.184.193
     tcp flags:!0x16/0x02
DROP       tcp  --  0.0.0.0/0            0.0.0.0/0
DROP       udp  --  0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Si todo está correcto, guardamos el estado de las reglas actuales, y configuramos iptables para que se ejecute al inicio:

```
minimum root # /etc/init.d/iptables save
minimum root # rc-update add iptables default
```

El funcionamiento de este cortafuegos es muy básico. Simplemente nos aseguramos de que sólo usuarios de subredes autorizadas de la universidad (la red de servidores y la de trabajadores de Servicios Centrales) tienen acceso a la administración de servicios que existirán en este PC. Esto incluye tanto SSH como los interfaces telnet de configuración de Zebra y RIP, que veremos más adelante. Prohibimos el acceso a estos servicios también a los usuarios de la VPN. Permitimos el acceso al demonio del protocolo RIP y al Radius delegado, desde las direcciones IP necesarias. Permitimos la entrada de datagramas IP conteniendo paquetes con el protocolo ESP, que se corresponderán con tráfico IPSec, desde las direcciones legítimas de los terminadores de túneles de Telefónica. Permitimos por último el tráfico al puerto 500 UDP desde dichos terminadores, utilizado para las transacciones del protocolo IKE. Por último, denegamos el resto del tráfico en su totalidad.

3.2.2. Servicios requeridos

Una vez tenemos un sistema seguro y configurado correctamente para actuar como un servidor, procedemos a instalar y configurar el software necesario para nuestros propósitos:

```
minimum root # emerge openswan quagga gnuradius
```

OpenSWAN

La configuración de IPSec tiene dos partes bien diferenciadas. Por un lado, la configuración de los parámetros generales y las VPN que vamos a establecer, y por otro la configuración de criptografía que compartimos con los terminadores de túneles de Telefónica. Los ficheros de configuración se encuentran en el directorio `/etc/ipsec/`.

Uno de los requisitos impuestos por Telefónica es la utilización de claves alfanúmericas de 8 caracteres, en mayúsculas, como secretos pre-compartidos para el intercambio de claves IKE. Como hemos visto en la descripción teórica de la introducción, esto es nefasto si se combina con el uso de IKE en modo agresivo. Para indicar a IPSec los secretos compartidos con los clientes remotos utilizamos el fichero `/etc/ipsec/ipsec.secrets`. En este fichero indicamos por cada línea un cliente conocido con el que compartimos un secreto, en el formato `IP_Local IP_Remota : PSK "SECRETO"`. Seguimos esta sintaxis para indicar los dos terminadores de túneles de Telefónica:

```
193.147.184.193 195.55.47.10 : PSK "SECRETO!"
193.147.184.193 195.55.47.11 : PSK "SECRETO!"
```

La configuración general de IPSec y los perfiles para las asociaciones de seguridad se establece en el fichero `/etc/ipsec/ipsec.conf`. Está estructurado en secciones, cada una con sus correspondientes tuplas de configuración, en la forma `variable=valor`. El bloque *setup* establece las opciones generales de la pila de protocolos de IPSec, como por ejemplo, si utilizar o no las opciones de *NAT Traversal*, o el nivel de detalle de los registros de actividad. Dentro del bloque *%default* definiremos las opciones que deseamos sean comunes a cada asociación de seguridad, de forma que no tengamos que especificarlas por cada conexión que creemos. Por último, definimos los perfiles *cifrador1* y *cifrador2* que describen los parámetros concretos de las conexiones con los terminadores de túneles

de Telefónica. Nos limitaremos a indicar las direcciones IP de los mismos, y la ruta a un script que deberá ejecutarse al iniciar una asociación de seguridad con ellos.

```
version 2.0     # conforms to second version of ipsec.conf specification

# configuración básica
config setup
        interfaces="ipsec0=eth0"
        klipsdebug=none
        plutodebug=none
        plutoopts=%search
        pluto=%search
        uniqueids=yes
        nat_traversal=no

# valores por defecto para todas las conexiones
conn %default
        # modo transporte
        type=transport
        keyingtries=0
        disablearrivalcheck=no
        authby=secret
        pfs=no
        ikelifetime=10000s
        # algoritmos de cifrado y resumen
        esp=3des-sha1
        keylife=3600s
        # ip del terminador urjc
        left=193.147.184.193
        # gateway
        leftnexthop=193.147.184.1
        leftprotoport=0/0
        rightprotoport=0/0
        auto=start

# perfil para conexiones con cifrador 1
conn cifrador1
        # script de inicio de la conexión
        leftupdown=/etc/conf.d/network-scripts/ud_tun0
        right=195.55.47.10

# perfil para conexiones con cifrador 2
conn cifrador2
        # script de inicio de la conexión
        leftupdown=/etc/conf.d/network-scripts/ud_tun1
```

```
        right=195.55.47.11
```

En la configuración anterior indicamos dos scripts, `/etc/conf.d/network-scripts/ud_tun0` y `/etc/conf.d/network-scripts/ud_tun1`, que se ejecutarán cuando se crean o se destruyen las asociaciones definidas en el perfil correspondiente. A continuación incluímos cada uno de los dos scripts referenciados, respectivamente. Cuando la asociación cambie de estado entre los terminadores, se eliminará el túnel en caso de existir, y se volverá a crear con los parámetros adecuados. Para que estos cambios se vean reflejados en el encaminamiento, reiniciaremos los demonios *zebra* y *RIP*.

```
# script que se ejecuta cuando la asociación ipsec cambia de estado
TUNEL=‘/sbin/ifconfig | grep tunnel0 | awk ’{print $1}’‘

if [ $TUNNEL ]; then
        ifconfig tunnel0 down
        ip tunnel del name tunnel0
fi
ip tunnel add name tunnel0 mode ipip remote 195.55.47.10 local 193.147.184.193 \
 dev ipsec0 ttl 255
ifconfig tunnel0 up 195.55.47.2 netmask 255.255.255.252 pointopoint 195.55.47.1 \
 mtu 1500 multicast

/etc/init.d/zebra restart
/etc/init.d/ripd restart


# script que se ejecuta cuando la asociación ipsec cambia de estado
TUNEL=‘/sbin/ifconfig | grep tunnel1 | awk ’{print $1}’‘

if [ $TUNNEL ]; then
        ifconfig tunnel1 down
        ip tunnel del name tunnel1
fi
ip tunnel add name tunnel1 mode ipip remote 195.55.47.11 local 193.147.184.193 \
 dev ipsec0 ttl 255
ifconfig tunnel1 up 195.55.47.6 netmask 255.255.255.252 pointopoint 195.55.47.5 \
 mtu 1500 multicast

/etc/init.d/zebra restart
/etc/init.d/ripd restart
```

Quagga

Pasamos a configurar las opciones necesarias para utilizar el protoclo RIP de encaminamiento de forma correcta, tal como requiere Telefónica. La configuración del software quagga se realiza mediante varios ficheros situados en `/etc/quagga/`. La configuración general de zebra se encuentra en el archivo `/etc/quagga/zebra.conf`. En ella definiremos los parámetros del *router* principal:

```
! nombre del router virtual
hostname minimum
! clave de acceso telnet al router virtual
password clave_acceso
! clave de acceso para modo privilegiado
enable password clave_admin
!
interface lo
interface eth0
interface ipsec0
!
interface tunnel0
 description SERVICIO MOVISTAR INTRANET
 multicast
!
interface tunnel1
 description SERVICIO MOVISTAR INTRANET
 multicast
!
line vty
```

Una vez configurado de esta forma, podemos arrancar el servicio del demonio zebra y añadirlo al inicio de la forma habitual. Podremos conectar por telnet al puerto 2601, lo que nos permitirá manejar el servicio de una forma similar a la de un router Cisco.

```
minimum root # /etc/init.d/zebra start
 * Starting zebra...                                                [ ok ]
minimum root # rc-update add zebra default
 * zebra added to runlevel default
 * Caching service dependencies...
 * rc-update complete.
```

Configuramos ahora las opciones concretas del router virtual RIP, en su archivo correspondiente, `/etc/quagga/ripd.conf`:

```
! nombre del router virtual
hostname minimum-rip
! clave de acceso telnet al router virtual
password clave_acceso
! clave de acceso para modo privilegiado
enable password clave_admin
!
interface lo
interface eth0
interface ipsec0
interface tunnel0
interface tunnel1
!
router rip
 timers basic 30 90 5
 default-information originate
 redistribute kernel
 network tunnel0
 network tunnel1
 distribute-list 1 out
! indicamos que esta será la ruta por defecto
access-list 1 permit 0.0.0.0
access-list 1 deny any
!
line vty
```

Realizamos el mismo proceso que con zebra. En esta ocasión, podremos acceder al interfaz tipo Cisco mediante telnet al puerto 2602.

```
minimum root # /etc/init.d/ripd start
 * Starting ripd...                                                    [ ok ]
minimum root # rc-update add ripd default
 * ripd added to runlevel default
 * Caching service dependencies...
 * rc-update complete.
```

GNU Radius

Por último, pasamos a dejar a punto nuestro servidor proxy de Radius, que se utilizará para realizar las consultas de autenticación del servicio. Sus ficheros de configuración se encuentran en el directorio `/etc/raddb/`. Modificaremos las opciones generales del servidor

en `/etc/raddb/config`. Para una información detallada sobre todas las posibles opciones de configuración se recomienda mirar la página de manual o ejecutar `info Radius config`. La mayoría de opciones por defecto serán más que suficientes.

```
# opciones generales
option {
        max-requests 1024;
        # usuario con el que ejecutará el demonio
        radiusd-user radiusd;
        # caracteres válidos en nombres de usuario
        username-chars ".-_";
        # resolución inversa de DNS
        resolve no;
};
# opciones de registro de actividad
logging {
        prefix-hook "default_log_prefix";
        channel default {
                file "radius.log";
                print-category yes;
                print-level yes;
        };
        channel info {
                file "radius.info";
                print-pid yes;
        };
        channel debug {
                file "radius.debug";
        };
        category auth {
                print-auth yes;
                print-failed-pass yes;
        };
        category info {
                channel info;
        };
        category =debug {
                channel debug;
        };
        category * {
                channel default;
        };
};
# opciones de autenticación
auth {
```

```
        max-requests 48;
        request-cleanup-delay 2;
        detail yes;
        # divide en dos los nombres de usuario, el
        # propio nombre de usuario y su realm
        strip-names yes;
        checkrad-assume-logged yes;
};
# opciones para las cuentas de usuario
acct {
        max-requests 48;
        request-cleanup-delay 2;
};
# configuración del módulo rewrite
rewrite {
        load "checknas.rw";
        load "log-hook.rw";
};
```

En el archivo de configuración `/etc/raddb/clients` indicaremos una lista de los servidores Radius a los que permitiremos realizar consultas a nuestro proxy, incluyendo el secreto compartido a utilizar con los mismos. Especificamos las direcciones IP de los servidores Radius de Telefónica:

```
194.224.26.133          SECRETO!
194.224.26.134          SECRETO!
```

Por contra, en el archivo `/etc/raddb/client.conf` escribiremos los datos concretos que deberá utilizar nuestro proxy para escalar sus consultas a los servidores Radius de la universidad.

```
server ganimedes 193.147.184.3 s3cr3t0 1812 1813
server europa 193.147.184.22 s3cr3t0 1812 1813
# dirección del proxy
source_ip 193.147.184.193
# tiempo a esperar por las respuestas
timeout 5
# número de reintentos
retry 1
```

Llegados a este punto tenemos configurada la red Radius de clientes y servidores que vamos a utilizar. Nos falta tan sólo para acabar con nuestras tareas de configuración indicar a nuestro proxy cual es el realm que debe escalar a los servidores privados, y qué debe hacer concretamente. Esto lo podemos realizar en el propio archivo `/etc/raddb/realms`:

```
# indicamos que los usuarios del dominio urjc sean autenticados por
# radius.urjc.es, que se divida el nombre de usuario y el realm en
# la consulta, y que no se tenga en cuenta la distinción entre
# mayúsculas y minúsculas
urjc            radius.urjc.es          ignorecase,strip
```

3.3. Análisis del sistema

Es importante no perder de vista el funcionamiento general del sistema que hemos construido, y tener siempre presentes las herramientas a nuestra disposición, para probarlo y solucionar los problemas que surjan sobre la marcha.

3.3.1. Monitorización de servicios

No debemos olvidar configurar un sistema de logs como *syslog* para que guarde registros de toda la actividad de nuestros servicios en alguna parte. En particular nos será especialmente útil para trazar el funcionamiento de IPSec. Por lo general los logs de IPSec se situarán en `/var/log/secure`. Podemos, por ejemplo, ver en tiempo real los mensajes que se generen según se realice la negociación IKE, de forma muy sencilla:

```
minimum root # tail -f /var/log/secure
```

El propio IPSec proporciona comandos muy útiles a la hora de depurar su funcionamiento. Por ejemplo, con `ipsec auto --status` podremos comprobar el correcto establecimiento de las sesiones IKE y los parámetros asociados a los túneles IPSec que se establezcan con los terminadores de Telefónica. Otra herramienta muy útil es `ipsec barf`, que genera gran cantidad de información de todo tipo relativa a la configuración de openswan y del sistema, así como un log del proceso de negociación y establecimiento de los túneles IPSec. Por lo general no es posible tratar toda la información que proporciona directamente desde la pantalla, por lo que conviene redirigir su salida a un fichero para su posterior análisis: `ipsec barf >ipsec.log`.

3.3.2. Funcionamiento general

Es momento de echar la vista atrás y revisar al detalle el funcionamiento del sistema en el que hemos integrado nuestro terminador de túneles IPSec. Al fin y al cabo, como ya dijimos en un primer momento, ésta no es más que una pequeña parte de una compleja infraestructura que nos permite ofrecer un servicio de conectividad a la red corporativa de forma ubícua, utilizando telefonía móvil. Observemos paso a paso el funcionamiento del sistema.

Acceso mediante GSM

Una vez disponible la infraestructura necesaria por parte de Telefónica y la Universidad Rey Juan Carlos para ofrecer el servicio, los usuarios tan sólo necesitan disponer de un PC conectado a un teléfono móvil GSM configurado con el software provisto por TME para realizar llamadas de datos.

1. En primer lugar el usuario marcará el número corto 553 (o el número largo +34 629 000 553), obteniendo así acceso al servicio de conexión a Internet de Telefónica.

2. Los RAS[5] de acceso GSM de Telefónica consultarán su Radius corporativo para crear un túnel.

3. El servidor Radius corporativo de Telefónica examinará el nombre de usuario que se desea autenticar y su *realm* asociado. Si éste coincide con el indicado por la universidad, *@urjc*, el servidor indicará al RAS un *router adaptador de túneles* concreto contra el que crear el túnel.

4. Una vez creado dicho túnel, será el propio router adaptador el que envíe una petición Radius al servidor de Telefónica para autenticar al usuario.

5. Nuevamente el Radius corporativo de TME comprobará el *realm* o *mnemónico* del usuario. En caso de pertenecer a la URJC escala la petición de autenticación al Radius instalado en el terminador de túneles de la universidad, a través de una infraestructura de túneles que finaliza en éste.

[5] *Remote Access Service.*

6. El Radius delegado de la universidad comprueba el realm del usuario que pide ser autenticado. Al ser reconocido, escala a su vez la petición a uno de los servidores Radius de la red de la universidad.

7. Los Radius internos de la universidad autentican al usuario contra el directorio activo LDAP[6] e informan de su decisión al Radius delegado.

8. El Radius delegado de la universidad responde adecuadamente al servidor Radius de TME que le hizo la petición.

9. El servidor Radius de TME dará al usuario por autenticado e informará al router adaptador y a un servicio de DHCP[7], que asignará al usuario una dirección IP del *pool* GSM determinado por la universidad.

10. A partir de este momento, todo el tráfico generado por el usuario es encaminado hacia la red de la universidad por los túneles correspondientes a la misma (cada corporación goza de túneles independientes), llegando así hasta el terminador de túneles IPSec de la misma.

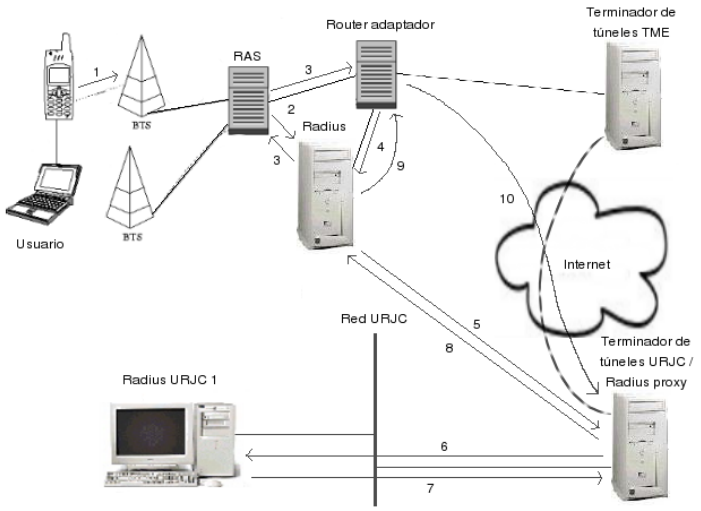

Figura 3.3: Secuencia de acceso para terminales GSM.

[6] *Lightweight Directory Access Protocol*. Se trata de un protocolo para acceder a servicios de *directorio* en línea.

[7] *Dynamic Host Configuration Protocol*. Protocolo para la configuración de dispositivos de red de forma dinámica según se conectan a la red.

Acceso mediante GPRS

Los requisitos para dar acceso remoto seguro a usuarios de terminales móviles con soporte GPRS son exactamente los mismos que para usuarios de GSM. De hecho, de cara al usuario lo único que cambia es el propio terminal y la forma de conectarse al servicio, que nuevamente estará indicada por Telefónica.

La secuencia de acciones que se realizan desde que el usuario accede a la red con su terminal telefónico hasta que se le da acceso a la red corporativa es muy similar a la ya vista con GSM. Sin embargo, existen ciertas diferencias que merece la pena poner de manifiesto:

1. En GPRS no se realizan llamadas de datos como en GSM. En su lugar, se establece un *contexto PDP*, que es básicamente el conjunto de información relativo a una conexión GPRS entre el terminal y la red de acceso.

2. No se utiliza un RAS para garantizar acceso al usuario a la red. Por contra, se utiliza conmutación de paquetes, y el acceso a la red corporativa se realiza a través de un GGSN[8], que es el nodo de la red que hace funciones de encaminamiento.

3. Mientras que en GSM se da el paso hacia tarificación por datos, gracias al mantenimiento de sesiones virtuales en el RAS, es en GPRS cuando se completa este proceso y se factura única y exclusivamente por volumen de datos intercambiado a través del servicio, permitiendo a los usuarios estar permanentemente conectados a su red corporativa sin cargo adicional.

3.3.3. Cuestiones de seguridad

Uno de los aspectos más importantes a lo largo del desarrollo de este proyecto ha sido la seguridad de las comunicaciones, los usuarios y los equipos que forman parte del mismo. Hasta el momento hemos analizado los posibles puntos débiles de nuestro trabajo y razonado elecciones que nos proporcionen un grado de seguridad lo más alto posible. Pero la realidad es que ningún sistema es seguro al cien por cien, y por muy restrictivos que seamos en nuestras reglas de filtrado, por mucho que utilicemos algoritmos y protocolos seguros, nunca tenemos garantizado el éxito en esta compleja tarea. Por eso la mejor forma

[8] *Gateway GPRS Support Node.*

de evaluar la seguridad de un sistema informático es intentar romperla uno mismo, antes de que lo hagan otros.

Existen diversas herramientas de libre distribución diseñadas para identificar e incluso explotar vulnerabilidades en entornos que hacen uso de IPSec. A continuación exponemos los programas cuyo uso se ha planificado para comprobar la seguridad de nuestro sistema. Algunas de ellas no han podido ser utilizadas hasta la fecha, ya que dependen de que se establezcan definitivamente sesiones de pruebas del servicio con Telefónica.

NMAP

Se trata de una herramienta multipropósito que sirve desde para identificar aplicaciones en una red a través de sus puertos conocidos, tanto TCP como UDP, hasta para obtener una estimación del sistema operativo que puede estar ejecutando una máquina conectada a la red.

En nuestro caso podemos utilizarla para identificar el servicio de intercambio de claves ISAKMP, comprobando si el puerto 500 UDP se encuentra abierto. Podemos encontrar esta herramienta además de una excelente documentación en su sitio web: `http://www.insecure.org/`.

Por ejemplo, si utilizamos `nmap` desde una máquina en la que no confiamos (con una dirección IP de origen que no pertenezca a los rangos permitidos de la universidad) para comprobar los puertos TCP abiertos, obtendremos un resultado similar al este:

```
deadstar root # nmap -sS -O 193.147.184.193

Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) at 2005-02-08 10:57 CET
Warning:  OS detection will be MUCH less reliable because we did not find at
 least 1 open and 1 closed TCP port
All 1659 scanned ports on 193.147.184.193 are: filtered
Too many fingerprints match this host to give specific OS details

Nmap run completed -- 1 IP address (1 host up) scanned in 88.580 seconds
```

Si por contra lo utilizamos para buscar puertos UDP abiertos, obtendremos como resultado que todos los puertos UDP aparecen abiertos. Esto es debido a que los paquetes UDP son filtrados a la entrada en el terminador de túneles, y por tanto no provocan ningún

paquete *ICMP* de *destino no alcanzable*. En tal caso, `nmap` asume que los puertos están abiertos, provocando falsos positivos.

IPSec Scan

Una herramienta para sistemas operativos Windows que permite comprobar si una IP o un rango de ellas tienen un servicio IPSec activo y accesible, con un grado de fiabilidad muy alto. La herramienta y un conjunto de preguntas más frecuentes se encuentra en `http://ntsecurity.nu/toolbox/ipsecscan/`.

La utilización de esta herramienta sobre nuestro terminador de túneles desde una dirección IP no fiable da un resultado negativo, al ser filtrado por completo el tráfico ESP.

IKE Scan

Un programa muy útil desarrollado por Roy Hill que registra las *huellas* de un servicio de intercambio de claves IKE, y las analiza para tratar de averiguar a qué software se corresponden. Éste puede ser un primer paso para tratar de vulnerar la seguridad de un servicio IPSec, ya que conociendo qué software se encuentra detrás del mismo, podemos buscar en sitios web que publican listas de vulnerabilidades alguna que afecte al software al que nos enfrentamos. `ike-scan` se encuentra disponible en `http://www.nta-monitor.com/ike-scan/`.

Un atacante que consiga utilizar esta herramienta contra nuestro terminador de túneles desde una dirección IP reconocida por el servicio IPSec podría obtener un indicio concreto y muy efectivo de la identidad del software que implementa IPSec en este PC. En concreto, la identificación positiva establece que se trata de un equipo ejecutando el sistema operativo Linux y el software para VPN FreeS/WAN. Sin embargo, para eso debería introducirse previamente en una de esas máquinas reconocidas. En caso contrario, el resultado que obtendría no le proporcionaría información alguna:

```
fury ike-scan-1.7 # ike-scan --showbackoff 193.147.184.193
Starting ike-scan 1.7 with 1 hosts (http://www.nta-monitor.com/ike-scan/)

Ending ike-scan 1.7: 1 hosts scanned in 2.499 seconds (0.40 hosts/sec).
0 returned handshake; 0 returned notify
```

IKE Probe e IKE Crack

Ya hemos avisado anteriormente de la problemática de usar claves pre-compartidas con el modo agresivo de intercacambio de claves en IKE. Advertimos que existían herramientas capaces de comprometer la seguridad de IPSec en este caso particular. `ikeprobe` es una de esas herramientas, creada por Michael Thumann y disponible en `http://www.ernw.de/download/ikeprobe.zip`. Con ella podremos obtener la clave de autenticación codificada que se usa en una asociación de seguridad concreta, para posteriormente romperla con algún software específico como `Cain & Abel` (`http://www.oxid.it/cain.html`).

`ikecrack` es el equivalente para entornos unix de `ikeprobe`, que se limita hasta el momento a claves MD5, aunque tiene planes para soportar en el futuro SHA1. Se puede encontrar en `http://ikecrack.sourceforge.net/`. Adicionalmente, en `http://www.ernw.de/download/pskattack.pdf` tenemos un excelente documento por Michael Thumann que explica de forma sencilla cómo atacar sistemas que hacen uso de claves pre-compartidas y el modo agresivo de IKE.

Utilizaremos estas herramientas para comprobar la seguridad de nuestras conexiones IPSec y las claves pre-compartidas utilizadas, a la hora de establecer el servicio entre los terminadores de túneles de Telefónica y nuestro propio terminador.

Capítulo 4

Conclusiones y trabajos futuros

Durante los últimos meses este proyecto ha supuesto una gran cantidad de trabajo, de estudio, investigación y pruebas, en campos en su mayoría no impartidos en esta titulación en la Universidad Rey Juan Carlos. Es por ello que son muchos los problemas que han surgido a lo largo de su desarrollo, y muchas las cosas aprendidas, como por ejemplo:

- Conceptos de tunelado y encapsulación en redes.
- Redes Privadas Virtuales.
- Software y protocolos para encaminamiento: Zebra y RIP.
- Seguridad a nivel de red: IPSec.
- Criptografía y algoritmos matemáticos.
- Gestión y configuración de cortafuegos con iptables.
- Configuración y administración de servidores y redes Linux.
- Arquitectura AAA (Autenticación, Autorización, Cuentas de usuario).
- El protocolo Radius.

Y por supuesto, siempre es grato trabajar en algo que realmente tiene una utilidad práctica y que nos podemos encontrar en cualquier sistema de comunicaciones real. En particular, resulta de especial interés el aprendizaje sobre sistemas de comunicaciones avanzados y estrechamente relacionados con el mundo de la seguridad informática, en continuo cambio y avance.

4.1. Evaluación del trabajo

Desde un principio hemos contado con unos requisitos bastante específicos, aunque a lo largo del proyecto cambiaron, aumentando la carga de trabajo necesaria. Esto, junto con la decisión del Servicio de Comunicaciones por obtener un servicio fiable y de calidad, que asegure tanto a nuestros usuarios como nuestra red ya existente, ha facilitado bastante la toma de decisiones.

A día de hoy hemos conseguido disponer de una maqueta muy segura y que no por ello deja de ofrecernos la flexibilidad requerida para los usuarios del servicio. El PC que hemos configurado durante este tiempo está bien protegido, al menos durante el tiempo que los algoritmos y protocolos que utiliza se consideren seguros, y gracias a ello no hemos necesitado introducir restricciones especiales sobre los usuarios de la VPN, que podrán disfrutar de un acceso a la red de la universidad desde cualquier parte y en unas condiciones óptimas.

4.2. Futuros trabajos

Pese a todo, esto ha sido tan sólo el principio de este trabajo. En adelante deberán realizarse extensas sesiones de pruebas concertadas con Telefónica para garantizar que los objetivos que enumeramos en el segundo capítulo se cumplen debidamente. Tras este periodo se instalará el terminador de túneles definitivo, bajo un sistema operativo Debian GNU/Linux, como ya se ha indicado, y con los servicios y configuraciones ya probados. Una vez se tenga certeza de que este servidor cumple correctamente su cometido, el servicio se considerará *en producción* y nuestros usuarios finales podrán comenzar a utilizarlo.

Más adelante, será interesante buscar formas de monitorizar y administrar el terminador de túneles de la forma más sencilla posible. Ya hemos visto algunas herramientas que simplifican estas tareas, pero sería interesate obtener cotas aún mayores de facilidad de uso. Se han pensado algunas aplicaciones para mejorar estos aspectos, y pruebas adicionales sobre la seguridad del sistema:

- Una aplicación para la administración del servicio de forma global (lo que incluye openswan, quagga y GNU/Radius como parte de un mismo servidor), de forma que activar y desactivar el servicio sea trivial, y cada parte del mismo se encuentre

monitorizada por si ocurriese cualquier tipo de problema.

- Por otro lado, un pequeño servidor web que permita ejecutar programas CGI que extiendan su funcionalidad y faciliten la consulta, mediante un simple navegador web, del estado del servidor y los registros del sistema.

- Para terminar de corroborar la seguridad del servicio, durante las sesiones de pruebas necesarias concertadas con Telefónica se utilizarán las herramientas descritas en el tercer capítulo desde todos los escenarios posibles (Internet, la VPN o los propios terminadores de túneles de Telefónica), para confirmar que la seguridad es la esperada.

Una vez llegados a este punto podremos dar nuestro trabajo por finalizado.

Bibliografía

[1] Página de *GNU Radius*. http://www.gnu.org/software/radius/.

[2] Página de *OpenSWAN*. http://www.openswan.org.

[3] Página de *Quagga Routing Suite*. http://www.quagga.net/.

[4] Página de *StrongSWAN*. http://www.strongswan.org.

[5] Página del proyecto *OpenVPN*. http://www.openvpn.net.

[6] Página del *VPN Consortium*. http://www.vpnc.org.

[7] Network Working Group. *RFC 1631 - The IP Network Address Translator (NAT)*. http://www.faqs.org/rfcs/rfc1631.html.

[8] Network Working Group. *RFC 2138 - Remote Authentication Dial In User Service (RADIUS)*. http://www.faqs.org/rfcs/rfc2138.html.

[9] Network Working Group. *RFC 2402 - IP Authentication Header*. http://www.faqs.org/rfcs/rfc2402.html.

[10] Network Working Group. *RFC 2406 - IP Encapsulating Security Payload (ESP)*. http://www.faqs.org/rfcs/rfc2406.html.

[11] Network Working Group. *RFC 2409 - The Internet Key Exchange (IKE)*. http://www.faqs.org/rfcs/rfc2409.html.

[12] Network Working Group. *RFC 3715 - IPsec-Network Address Translation (NAT) Compatibility Requirements*. http://www.faqs.org/rfcs/rfc3715.html.

[13] Jonathan Hassell. *Radius*. O'Reilly, 2002. Excelente libro relativo a los protocolos AAA y enfocado a Radius y su implementación libre FreeRadius.

[14] Elizabeth Kaufman. *Implementing IPSec*. Wiley, 1999. Completo estudio de la arquitectura de IPSec y su implementación.

[15] Chris McNab. *Seguridad de Redes*. Anaya Multimedia / O'Reilly, 2004. Técnicas y herramientas para la evaluación de seguridad en redes.

[16] The RFC-Editor. *Official Internet Protocol Standards List*. `http://www.rfc-editor.org/rfcxx00.html`.

[17] The RFC-Editor. *RFC Frequently Asked Questions*. `http://www.rfc-editor.org/rfcfaq.html`.

[18] William Stallings. *Comunicaciones y Redes de Computadores*. Prentice Hall, sixth edition, 2000. Fundamentos técnicos de las redes de computadores.

[19] James S. Tiller. *A Technical Guide to IPSec Virtual Private Networks*. Auerbach, 2001. Completa descripción técnica de IPSec y sus peculiaridades para el diseño e implementación de redes privadas virtuales.