INAP
INSTITUTO NACIONAL DE ADMINISTRACIÓN PÚBLICA

Curso *online*: **Seguridad en Redes WAN e Internet**

Módulo 5 SEGURIDAD EN INTERNET

Autores: Daniel Díaz y Andrés Marín

Índice de contenidos

Capítulo 1 INTRODUCCIÓN A LA SEGURIDAD EN INTERNET

El término Internet sirve para designar la interconexión de miles de servidores TCP/IP. Internet es la Red de redes que interconecta redes alrededor de todo el mundo facilitando:

- Acceso a un número casi ilimitado de recursos: sistemas, fuentes de información, bases de datos, ficheros y datos de carácter público, foros de discusión y de noticias, etc. A un conjunto de usuarios cada vez más conectados y móviles.
- Posibilidad de nuevas formas de interactuar como las administraciones electrónicas y nuevas oportunidades de negocio: comercio electrónico, teletrabajo, difusión y distribución de información, etc.
- Integración de los sistemas de información de la organización, y la conexión de diversos dispositivos en la Internet de las cosas, ahorrando costes de infraestructura de comunicaciones.

La principal ventaja de Internet, y una de las razones principales de su crecimiento vertiginoso es su carácter abierto; a la vez, esta característica también supone uno de sus principales inconvenientes en relación con la seguridad de la información. La otra cara de la moneda del mundo Internet es la seguridad, se trata de balancear las ventajas y los riesgos de participar en Internet.

Al igual que en cualquier sociedad, en Internet existe un pequeño porcentaje de personas maliciosas, y aunque ese porcentaje sea menor del uno por ciento del total de internautas, como los usuarios de Internet se cuentan por cientos o miles de millones, el número potencial de atacantes es lo suficiente grande como para que deba preocuparnos.

Internet fue creada para el libre acceso a la información, regida principalmente por políticas de buen uso de la red, pero partir de 1990, cuando comienza la integración de la actividad comercial y económica en Internet, la seguridad pasa a ser una de las principales preocupaciones.

En este módulo vamos a repasar algunas de las aplicaciones y servicios más populares de Internet junto con alguna de sus amenazas para finalmente exponer los protocolos más comunes que utilizan para proporcionarles seguridad.

1.1. SERVICIOS Y AMENAZAS EN INTERNET

La mayoría de los servicios de Internet se basan en el **modelo cliente/servidor,** donde un programa solicita servicios de otro. Al programa que solicita el servicio se le llama cliente y al que responde a la solicitud servidor, ambos programas pueden ejecutarse en el mismo ordenador o, lo que es más común, en ordenadores diferentes. Según el tipo de servicio prestado, se puede hablar, por ejemplo, de servidores de correo, de ficheros, de noticias o servidores Web.

Los programas cliente suelen ejecutarse en ordenadores y dispositivos personales y los **programas servidores** tienden a ser ejecutados en ordenadores más potentes conectados de forma permanente a Internetque pueden atender a muchos usuarios al mismo tiempo. Sin embargo, esta distinción no es muy útil, ya que existen tanto clientes como servidores para todo tipo de máquinas y sistemas operativos.

Una misma **máquina servidora puede albergar muchos servicios a la vez** y no es necesario emplear una máquina diferente para cada uno de ellos. En realidad cada servicio lo proporciona un programa que está corriendo continuamente en la máquina servidora y que está escuchando permanentemente la red en espera de peticiones de los clientes. Dichos programas se denominaban daemons (demonios) en los tiempos iniciales de UNIX. Por ejemplo, para el servicio del World Wide Web: en los servidores se ejecuta algún programa del tipo Internet Information Server de Microsoft, Apache, etc. E incluso combinados o por encima de éstos podemos tener otras aplicaciones como un sistema gestor de contenidos (CMS) tipo joomla, o una herramienta de distribución de tareas tipo hadoop, etc.

Internet suministra a los usuarios varios **servicios básicos** como:

- Servicio multimedia de información: World Wide Web (**HTTP**), y servicios transportados por éste:
  - Servicios de mensajería instantánea,
  - Servicios de presencia,
  - Redes sociales.
- Correo electrónico (**SMTP**).
- Transferencia de archivos (**FTP**).
- Acceso de terminal remoto (**Telnet**).
- Búsqueda de nombre de anfitrión/dirección (**DNS**)

El logro fundamental de Internet es haber englobado bajo una misma herramienta a varios servicios. Desde un navegador se puede **acceder a distintos servicios** sin necesidad de cambiar de entorno o utilizar programas separados. Hasta tal punto el sistema Web es globalizador de servicios que incluso las direcciones para acceder a las máquinas servidoras se unifican en una sola notación denominada **URL** (Universal Resource Locutor) Localizador Universal de Recursos.

Mediante las **direcciones URL**, es posible acceder desde un navegador a cualquier tipo de servidor de recursos, sea FTP, Telnet, news, correo, etc. y también por supuesto a un servidor de Web. Con la URL se identifica el recurso, su ubicación en la red (nombre de máquina y

directorio) y la forma de acceso (el protocolo a usar), su **formato genérico** para los servicios principales de Internet es el siguiente:

Servicio WWW, **http://host:port/path**

Servicio Telnet, **telnet://login:password@host:port** , o **telnet://host:port**

Servicio de FTP, **ftp://login:password@host:port:path** o **ftp://port/path**

Servicio de correo, **mailto:mailbox@domain**

Donde host es el **nombre de máquina** o la **dirección IP**. El número de puerto suele omitirse si el recurso se halla en un puerto estándar, que es lo habitual (23 para Telnet, 20 para FTP u 80 para http (WWW). Por ejemplo, si en las direcciones WWW, se omite el número de puerto, el formato de la URL queda:

http://host/path

Ninguno de los servicios citados **es** en realidad **seguro**, cada uno tiene sus debilidades. Antes de decidir soportar un servicio, se debe evaluar lo importante que es para sus usuarios y si se podrá protegerles de los peligros que subyacen a su implantación. Hay varias formas de hacerlo: ejecutar los servicios sólo en ciertas máquinas protegidas, emplear variaciones especialmente seguras de los servicios estándar; o, en algunos casos, bloquear los servicios por completo desde o hacia algunos o todos los sistemas externos. Es fundamental la correcta instalación y configuración del servidor, su actualización continua y una información constante sobre cómo proteger nuestro servidor de nuevas técnicas de intrusión.

Aparte de los servicios destinados al usuario final, también existen otros tipos de servicios y sus correspondientes servidores destinados a tareas de administración de la Intranet respectiva, a tareas de acceso a Internet, o de control de la seguridad, tales como **servidores de nombres**, **servidores proxy** o **DHCP**, etc., que sólo interesan a los administradores de redes.

A continuación se resumen la mayoría de los **servicios que se suelen suministrar a través de Internet**, así como algunas consideraciones sobre la seguridad o inseguridad que comporta su implantación:

1.2. ACCESO AL World Wide Web (WWW)

El **World Wide Web** (WWW, o la web (telaraña)) es un concepto que nace con Internet, basado en servicios ya existentes y en un protocolo nuevo: el **Protocolo de Transferencia de Hipertexto** (HTTP o HyperText Transfer Protocol).

La Web utiliza tecnología de **hipertexto** para enlazar y visualizar por la pantalla del cliente diversos tipos de documentos: texto, imágenes, sonido, video y otros formatos. El hipertexto permite ir de un documento a otro en Internet, “navegar”, los usuarios pueden moverse de uno a otro, sin importar en dónde estén almacenados, con solo hacer clic en una palabra o imagen para la cual ha sido definido un enlace.

La utilidad del Web se basa, en gran medida, en su flexibilidad, pero ésta dificulta su control. Los navegadores de Web se han convertido en un interfaz ubicuo que permite a los usuarios acceder y utilizar la inmensa mayoría de las aplicaciones en red (incluso en local). Esto se hace cada vez más patente, y lo vemos refrendado por la extensión de la computación en la nube (cloud computing) que sitúan nuestros datos y aplicaciones en servidores distribuidos y de máxima fiabilidad (24x7) de forma que utilizando un navegador y una conexión a Internet podamos acceder y manipular nuestras fotografías, documentos, presentaciones, vídeos, paneles de control de nuestra organización, etc. La gran popularidad de los teléfonos inteligentes y el abaratamiento de las tarifas planas de datos en España es otro motivo del auge de la computación en la nube. Una de las cosas más **peligrosas** que se pueden hacer con un dispositivo conectado a Internet es **descargar un programa y ejecutarlo**.

Al descargar y ejecutar un programa nos ponemos, por completo, en manos del autor del mismo.

Un servidor web confiable, **debe asegurar**:

- **El servidor y los datos que contiene**. Es necesario asegurarse de que el servidor pueda continuar operando, que la información que reside en él no sea modificada sin autorización y que sea distribuida solo a quienes se desea distribuir.
- **La información que viaja entre el servidor web y el usuario**. La información que proporciona el usuario al servidor web (nombre de usuario, claves de acceso, información financiera, etcétera) no debe poderse leer, modificar ni destruir por terceros. Se suele utilizar el término “servidor web con facilidades criptográficas”, en vez de “servidor web seguro”, para denominar a un servidor web que instrumenta protocolos criptográficos
- **Los sistemas de los usuarios**. Es necesario garantizar que la información, datos o programas descargados en los sistemas de los usuarios no ocasionarán daños.

Asegurar un servidor web conlleva tres **actuaciones**:

- **Garantizar que los usuarios autorizados para usar el sistema tengan las capacidades necesarias para hacer su trabajo y solo esas capacidades**.

Es importante controlar los privilegios de ejecución del servidor; no es conveniente que el servidor se ejecute con privilegios de root/Administrador, pues en ese caso, si un intruso consiguiera acceder al control del servidor, podría ejecutar cualquier comando del sistema.

Para obtener seguridad adicional sería necesario que el control se ejecutara en una partición independiente que no tuviera capacidad de modificar ninguna parte del navegador ni ningún otro ejecutable en el sistema operativo.

- **Restringir el acceso**. El servidor debe encontrarse en un sitio seguro, de forma que no haya acceso físico de personal no autorizado. Se debe limitar el número de usuarios que puedan iniciar sesiones interactivas con el ordenador. Siempre que sea posible, el servidor debe emplearse solo para un propósito. Además, se debe exigir que quienes acceden al servidor para su administración lo hagan utilizando un método seguro, como Telnet kerberizado, SecureID, S/KEY.

Hay que tener presente que las especificaciones de HTML y los navegadores actuales permiten acceder directamente desde el WWW, a otros servicios de Internet tales como FTP, TELNET, etc., es decir transportar otros protocolos utilizando HTTP. Esto puede ser utilizado para saltarse la protección que brindan determinados elementos de seguridad, como los cortafuegos o los encaminadores de protección, que realizan operaciones de filtrado en función del tipo de servicio.

- Utilizar aplicaciones web en el servidor seguras y asegurar la instalación de parches. Verificar regularmente la integridad de los contenidos publicados, los registros de acceso y los servidores backend (pro ejemplo bases de datos) utilizados en la aplicación web.

  Los servidores web son a menudo un medio valioso para los atacantes. Como fín en si mismo para acceder a los datos e incluso cambiarlos, y particularmente en las bases de datos con las que interactúan.

  Como puente para otros ataques también son interesantes pues pueden conseguir cargar código (generalmente javascript) maligno en el navegador, apoderarse de las cookies del usuario y suplantar su personalidad o apoderarse de sus datos, e incluso utilizar al navegador para que les dé información sobre posibles vulnerabilidades para a continuación explotarlas.

  Típicamente se utilizan en combinación con ataques de ingeniería social.

El proyecto OWASP (Open Web Application Security Project owasp.org) tiene abundante información sobre el tipo de vulnerabilidades que hay que proteger en las aplicaciones web, las más habituales son:

- La inyección de todo tipo de código generalmente sql, pero también puede ser ccs, svg, js, etc., para acceder a la base de datos desde el navegador, etc.
- El abuso de los sistemas de autenticación y gestión de sesiones
- La inyección de enlaces a scripts en otro sitio distinto (y generalmente malicioso), a esto se denomina cross-site scripting

OWASP publica anualmente un informe sobre las vulnerabilidades y ataques más frecuentes y dañinos que denomina “The Top 10 Project”. Están accesibles en owasp.org.

El principal **problema que se presenta para asegurar la información** que viaja entre el servidor web y el usuario, es que si no se usa TLS (ver siguiente sección) todos los datos se transmiten en claro. Esto unido a que ni el cliente ni el servidor tienen el control sobre la ruta que sigue la información que intercambian, ni sobre las máquinas que en el camino han podido acceder a ella, hace que no exista garantía sobre la seguridad de dicha información. Existen varias formas de proteger la información en tránsito:

- **Asegurar físicamente la red**, de forma que la intercepción sea imposible.
- **Ocultar la información** que se desea asegurar dentro de otra información que no tiene importancia.
- **Cifrar la información** de forma que no pueda ser descifrada por nadie que no posea la clave correcta.

De estas técnicas, la única práctica es el cifrado. Asegurar Internet físicamente es imposible, y ocultar la información sólo funciona si la gente de la que se oculta no sabe cómo está oculta. Hay una tendencia muy importante para utilizar cifrado en HTTP a través de la utilización de TLS en la actualidad (anteriormente de SSL pero ya está muy desaconsejado). De hecho la última versión de HTTP 2.0 incorpora TLS de forma obligatoria para algunos navegadores (h2).

Otro riesgo que corre la información en tránsito son los **ataques de denegación del servicio** resultante de interrupciones en la red. Una denegación del servicio **puede provenir de**:

- Un **evento físico**. P. e. el corte de un cable.
- Un **evento lógico**. P.e. un error en las tablas de enrutamiento de Internet.
- Un **ataque contra los servidores de una empresa**. P. e. un atacante puede bombardear al servidor web con millones de solicitudes por segundo, evitando que pasen peticiones legítimas. Los últimos ataques de denegación de servicio distribuido que hemos visto en Internet han movido hasta 1 Tbps () en este caso los ataques utilizaron 145.000 cámaras web infectadas.

Un ejemplo típico de inundación de los servidores web se deriva del funcionamiento del protocolo TCP/IP:

En este ataque, un cliente malicioso envía un mensaje de sincronización, SYN, que solicita el establecimiento de una nueva conexión con un cierto puerto del servidor e inicia el proceso de negociación de los parámetros de la conexión con el servidor. Este responde con mensajes SYN/ACK que sirven de confirmación de la recepción del primero y prosigue la negociación de los parámetros de la conexión. En este momento la conexión todavía se encuentra inconclusa y recibe el nombre de semiabierta.

En condiciones normales el cliente respondería con un mensaje ACK con el que confirmaría al servidor la recepción del SYN/ACK y la conclusión de la negociación de los parámetros de la conexión. A partir de este momento comenzaría el intercambio de información. Sin embargo, el cliente malicioso no responde al SYN/ACK con lo que el servidor, tras esperar un cierto tiempo prefijado, repetirá el ACK, y lo seguirá haciendo un número predeterminado de veces, doblando cada vez el tiempo de espera anterior con cada nuevo ACK. Finalmente después de 5 ó 6 mensajes ACK remitidos esperará un último lapso también determinado y cerrará la conexión semiabierta.

Si el atacante envía abundantes y reiteradas peticiones SYN sin responder a las respuestas SYN/ACK del servidor, éste irá abriendo y manteniendo sucesivas sesiones semiabiertas hasta que su capacidad de memoria prevista se desborde y lo haga caer.

Naturalmente, para que el servidor no pueda detectar al cliente, éste deberá falsificar su dirección IP, eligiendo en su lugar una inexistente o perteneciente a un cliente no activo, pues de otro modo éste podría responder a los sucesivos SYN/ACK del servidor. Este ataque se denomina SYN-flooding, o desbordamiento de SYN, y ataca el diseño del protocolo TCP, que reserva recursos del servidor antes de que la conexión esté establecida. Posiblemente la solución más extendida es la utilización de SYN-cookies, que se basan en devolver un número de ACK generado a partir de los datos de la solicitid de conexiñon inicial y un código solo

conocido por el servidor, de forma que el servidor no tiene que reservar ningún recurso hasta que reciba el ACK del cliente, si se verifica correctamente el ACK generado anteriormente por el servidor.

Asegurando el ordenador del usuario

El uso de archivos de HTML sencillos y los de imágenes por sí mismos no representan una amenaza directa a los usuarios (más allá de los problemas legales que pueden surgir de su contenido). Pero para aumentar las posibilidades de interacción con el Web, se promueven tecnologías como objetos **JavaScript (js), addons-extensiones-complementos** y **plug-ins** (conectores), que permiten a los desarrolladores "dar vida" a las páginas web y crear nuevos tipos de aplicaciones que no son posibles con formas de HTML sencillas. Cada vez están más en desuso el uso de Flash, los applets de Java (por problemas de seguridad fundamentalmente) y el uso de ActiveX.

Los creadores de Java intentaron limitar **efectos dañinos** de los applets. Para ello, hacen que estos pequeños programas se ejecuten en un espacio de memoria aparte, llamado "Sandbox" (cajón de arena). En la práctica, esto supone que **los applets sólo pueden acceder a recursos muy limitado**s dentro del cajón de arena. Para que puedan acceder al disco, se ha establecido un sistema de firmas digitales similar al ya establecido para **Active X**.

Aún a pesar de todo lo anterior, a través de Java se pueden sufrir diversos **tipos de ataques**: robo o destrucción de información, robo de recursos, denegación de servicio y los producidos por los virus.

Active X fué la respuesta de Microsoft a Java. Se trata de una versión reducida de OLE, es decir, son como los controles de Windows (controles de edición, botones listas, check box, etc.) pero más específicos y sofisticados pues permiten la descarga de pequeños objetos ejecutables que pueden ser llamados directamente desde la máquina del usuario. Esto evolucionó a .NET.

La seguridad tanto de Java como .NET está basada en **firmas digitales**. El software nos indica quién quiere descargar información en nuestro ordenador y si permitimos o no que lo haga. El problema es que queramos descargar un programa de alguien del que no sabemos nada. La mejor opción es no descargar nada a no ser que estemos totalmente seguros de la confianza que ofrece el emisor.

El programador de una aplicación debe pasar por una autoridad independiente de certificación que le dé el visto bueno. Para ello, la entidad certificadora atestigua la fiabilidad de la firma digital. Las entidades certificadoras únicamente certifican que empresa o desarrollador ha firmado el código, pero no compueban si contiene código maligno.

Los "plug-in" son trozos de código que se añaden a los "**browsers**" para obtener nuevos servicios. En algunos casos, como Java, son auténticos sistemas de programación, con sus correspondientes problemas. Se recomienda que es mejor utilizar "plug-in" que no ofrezcan funcionalidades de tipo general: es mejor que uno maneje imagen, que otro maneje vídeo, y

otro más sonido, en lugar de un sólo sistema para manejar todo tipo de animación. Otra buena práctica es no navegar por la Red desde ordenadores con información importante.

Dos "**plug-ins" populares** son: **Shockware de Macromedia**, para reproducir secuencias animadas y **Acrobat de Adobe**, que permite desplegar archivos PDF, aunque se comienzan a popularizar visores basados en javascript.

Una forma de mejorar la seguridad del código descargado es **confiar sólo en código de proveedores** que cuenten con una buena reputación y que sigan altos estándares de calidad al escribir sus programas.

Por último la forma más popular de software en el navegador actualmente son las denominadas extensiones (**addons**) que al igual que los plug-ins se integran perfectamente en el navegador , pero no como una biblioteca nativa sino a través de tecnologías puramente web como HTML, XML, CSS y JavaScript. Al igual que los plug-ins permiten particularizar totalmente la navegación al usuario pero igualmente representan un desafío de seguridad, pues una extensión maliciosa puede hacer lo que quiera sin conocimiento del usuario. También existen repositorios de complementos de confianza y la posibilidad de firmas electrónicas. NUNCA jamás instalaremos un addon que no venga firmado a nos ser que lo hayamos desarrollado nosotros. En Mozilla (addons.mozilla.org) se ofrece la posibilidad de recibir addons, analizarlos y si pasan el control se firman de forma automática. El Chrome store cobra 5$ para registrarse como desarrollador y registrar una pareja de claves que se utilizarán a lo largo de la vida del producto (para subir las nuevas versiones). Google asegura que hace análisis estático del código para detectar vulnerabilidades, aunque esto no siempre funciona.

1.3. CORREO ELECTRÓNICO

El correo electrónico permite enviar datos de un lugar a otro de forma similar a como lo hace el correo postal, es uno de los servicios de redes más populares y básicos, pero falsificar correo electrónico es sencillo, y las falsificaciones facilitan los ataques contra la reputación de las personas y la manipulación social.

El Protocolo Simple de Transferencia de Correo SMTP, (*Simple Mail Transfer Protocol)* es el protocolo estándar de Internet para enviar y recibir correo electrónico. SMTP en sí no es un problema de seguridad, pero lo pueden ser los servidores SMTP. Entre ellos el más común, en UNIX, es *Sendmail*.

La principal amenaza puede provenir de engañar al servidor de STMP para que, como parte del mensaje, ejecute una *shell script*. Esto permitirá a un atacante robar el archivo de "*passwords*", o abrir puertas falsas que posteriormente le permitirán hacer *Telnet*.

En la práctica los problemas más comunes con el correo electrónico son:

- Inundaciones inadvertidas (cadenas de cartas) o intencionadas (*spam*)
- Envío de datos confidenciales por medio del correo de Internet.

- Envío de programas que a su vez contengan caballos de Troya, virus o gusanos se recomienda revisar dichos programas antes de su ejecución.

SMTP tiene una serie de mejoras que permiten la autenticación de las conexiones (para autenticar los usuarios remitentes), para cifrado y firma de mensajes (S/MIME, PGP), para definir qué máquinas o rangos de direcciones son autorizados para enviar correo con origen en un determinado dominio (SPF), etc.

1.4. TRANSFERENCIA DE ARCHIVOS

El **Protocolo de Transferencia de Archivos, FTP** (File Transfer Protocol) es el protocolo estándar de Internet para este propósito. FTP es uno de los servicios más populares, ya que muchos internautas buscan y se bajan de la red, programas "shareware", actualizaciones de programas comerciales, demostraciones, documentos, etc. Los ficheros descargados pueden ser una vía de entrada de virus, máxime si se traen de sitios FTP sin garantías. Los ficheros descargados de la red, debemos almacenarlos en disco y analizarlos con el antivirus antes de ejecutarlos. Además debemos evitar la descarga de ficheros provenientes de páginas "underground".

En realidad, lo que más problemas genera, aunque no representa un riesgo importante de seguridad, es que los usuarios obtengan juegos de ordenadores, **software pirata** e imágenes pornográficas, que tienden a ocupar mucho espacio en disco y a través de ellos se pierde una cantidad significativa de tiempo. Por ello **se debe**:

- **Informar y mentalizar** a los usuarios para que desconfíen de cualquier software que obtengan por medio de FTP.
- **Comunicar** a los usuarios las **políticas establecidas** sobre acoso sexual y el uso de los recursos de la organización.

Hay un tipo de **FTP llamado anónimo** (anonymous FTP), que permite a los usuarios remotos acceder a archivos colocados en un área pública separada, sin dejarles iniciar una sesión y, potencialmente, tener acceso a todo el sistema. El área de FTP anónimo de un sitio puede ser un archivo público de documentos, software, imágenes o información de cualquier tipo que las personas necesiten o que se quiera compartir con ellas.

Para instalar un servidor FTP anónimo, **se debe asegurar** que las personas que lo vayan a utilizar no puedan tener acceso a otras áreas o archivos del sistema. También se deberá asegurar que los usuarios internos no utilicen el servidor de manera inapropiada. Puede ser tentador poner, en la zona anónima, archivos que se quiere que lean personas específicas, sin pensar que cualquiera en Internet puede leerlos, o sí lo hacen creen, ilusamente, en la seguridad a través de ser desconocido. Un buen planteamiento es tener una máquina dedicada exclusivamente a FTP (y otra a WWW).

Otros mecanismos para transferir archivos entre sistemas son scp y rsync. Están diseñados para utilizarlos a través de Internet, porque emplean un modelo de autenticación basado en la clave pública del anfitrión, y permiten la autenticación del usuario en la máquina remota a

través de contraseña o clave pública. Para poder utilizar estos comandos con garantías es necesario que el administrador garantice que las huellas digitales de las distintas máquinas que se almacenan en ficheros known_hosts sean las correctas, por ejemplo mediante su publicación en una página HTML protegida con HTTPS.

1.5. ACCESO DE TERMINAL REMOTA Y EJECUCIÓN DE COMANDOS

Permiten utilizar un **sistema remoto** como si fuera un terminal conectado directamente.

Telnet es el estándar para **acceso como terminal remoto en Internet**. Imita a un terminal no a un puesto de trabajo, sólo proporciona acceso a aplicaciones basadas en caracteres.

En tiempos, Telnet se consideró un servicio más o menos seguro porque requiere autenticación de usuario. Por desgracia, Telnet envía toda su información sin codificar, incluida la contraseña del usuario, lo que lo hace muy vulnerable a ataques de espionaje y robo. **Telnet es seguro sólo si la máquina remota y todas las redes entre ella y la máquina local son seguras**, lo cual significa que no es seguro a través de Internet.

Existen una versión segura de telnet (Telnets) que utiliza autenticación basada en clave pública de las máquinas y cifra la comunicación para asegurar que la contraseña (si se utiliza) no viaje en claro. Además existen otros programas, que pueden usarse para tener acceso como terminal remoto y **ejecución remota de programas** (**slogin, ssh, rdesktop**). Estos programas se pueden utilizar libremente en Internet, aunque siempre es necesario que el administrador vele por la correcta distribución de las claves públicas entre las máquinas que van a interactuar.

1.6. INFORMACIÓN SOBRE PERSONAS

Internet **no tiene un servicio** adecuado para **buscar información** sobre las personas en la red. Aun cuando, por ejemplo, se conozca el nombre real de una persona y dónde trabaja no se puede ir a un lugar central para buscar el nombre de usuario o la dirección de correo electrónico de esa persona. En el pasado se utilizaban dos servicios que proporcionaban cierta información sobre las personas: **finger y whois**. Hoy en día estos servicios no tienen ninguna relevancia más allá de redes internas unix.

El servicio whois es similar a finger, pero obtiene información disponible al público sobre anfitriones, redes, dominios y sus administradores. Por ejemplo en nic.esse puede acceder a los datos de los contactos técnicos de cualquier dominio registrado en el dominio "es". De hecho, es posible darse de alta para utilizar directamente el servicio whois a través del puerto 43. El servicio finger es un servicio restringido a redes unix y cada vez más en desuso.

El reemplazo de esto servicios ha venido de la mano del servicio de directorio: LDAP. LDAP es un protocolo estandarizado por la IETF junto con unos esquemas que facilita la obtención de información de personas de una organización. LDAP está integrado con los módulos de autenticación de los sistemas corporativos como GINA y Active Directory en Microsoft, o PAM en Unix, a fín de mantener la información de autenticación y autorización en un servicio independiente. Resulta muy eficaz a la hora de establecer y mantener dominios lógicos y controladores de dominios en sistemas distribuidos, juntamente con el servicio de DNS. Esto se puede hacer sin necesidad de equipamiento caro para establecer mecanismos de VLANs

complejas dentro de la organización, aunque puede redundar en un mejor servicio. El servicio de directorio permite tener una representación de usuarios de acuerdo con diferentes esquemas de datos estandarizados por la IETF, que incluyen la posibilidad de almacenar certificados X.509 de usuario.

En la actualidad, los buscadores y las redes sociales son el instrumento preferido por los usuarios de Internet para conocer información acerca de una persona. Existen redes sociales especializadas en relaciones laborales y profesionales, como linked-in, otras orientadas a nichos muy específicos como Tuenti o Research-gate, y otras de perfil amplio como Facebook o Google+. Estas redes recopilan gran cantidad de información de los usuarios y las relaciones con otros usuarios, construyendo perfiles con inmensa cantidad de datos que pueden poner en peligro la privacidad de los usuarios, dado que la utilización y la vigencia de dichos datos escapa al control de los usuarios, y en ocasión también a las propias redes sociales, quedando en un limbo almacenado en redes de distribución de contenidos. En cualquier caso es necesario que el usuario revise los controles de privacidad y opte por aplicar las mayores restricciones posibles a los datos de su perfil, de forma que retenga cierto poder sobre sus datos, y amplíe los permisos solo en los casos que sea imprescindible.

1.7. SERVICIOS DE CONFERENCIAS EN TIEMPO REAL

El correo electrónico y las noticias de Usenet están diseñadas **para facilitar comunicaciones asíncronas**; funcionan aunque los participantes no estén conectados en ese momento. Los servicios de conferencias en tiempo real, están diseñados para que los participantes los empleen de modo interactivo.

En los 90 aparecieron diversos servicios de conferencia y videoconferencia disponibles en Internet, tales como: talk, IRC, Multicast Backbone (MBONE).

En la actualidad, hay múltiples servicios de mensajería más sofisticados que se establecen a través de protocolos estandarizados como XMPP o SIP, que permiten orquestar conferencias y organizar canales UDP y TCP y transportar multimedia. También existen protocolos propietarios como el utilizado por Skype o LINE. En la actualidad hay una competición abierta en la que participan casi todos los actores imaginables para hacerse con el control de la mensajería y conferencias de los usuarios (skype, Messenger, Hangouts, Whatsapp, Telegram, Joyn, etc.), dada la importancia de los terminales móviles hoy en día.

1.8. SERVICIO DE NOMBRES

El servicio de nombres se encarga de **traducir los nombres de anfitrión** que utilizan las personas a las direcciones IP numéricas que utilizan las máquinas. Con millones de anfitriones enlazados, no resulta práctico para ningún sitio mantener una lista de anfitriones con el nombre y número de cada máquina en Internet y mucho menos que la tenga cada sitio. En lugar de eso, el **Servicio de Nombres de Dominio DNS** (Domain Name Service) permite que cada sitio tenga información sobre sus propios anfitriones y pueda encontrar la información para otros sitios.

DNS no es un servicio a nivel usuario en sí mismo, pero da soporte a los protocolos de aplicación como SMTP, FTP, Telnet y casi cualquier otro servicio que necesiten los usuarios. Por ejemplo se puede escribir "telnet ficticio.com" en lugar de "telnet 10.100.242.32". Además, muchos servidores de FTP anónimo no permiten conexiones de clientes a no ser que puedan utilizar un DNS para buscar el nombre de anfitrión del cliente a fin de iniciar la sesión.

El servicio de nombres se debe configurar para tener disponible la información completa para los anfitriones internos, pero sólo en forma parcial para los solicitantes externos.

Usar DNS interno y luego depender de los nombres de anfitrión para dar autenticación lo hace **vulnerable** a un intruso que puede instalar un servidor DNS mentiroso. Esto se puede manejar combinando algunos **métodos**, tales como:

Usar direcciones IP (en lugar de nombres de anfitrión) para dar autenticación a los servicios que deben ser más seguros.

Dar **autenticación a usuarios** en lugar de a anfitriones en los servicios más seguros, porque las direcciones IP también pueden falsificarse.

Los servidores raíz del servicio de nombres nunca han sucumbido a ningún ataque, aunque éstos se han producido. La mayor vulnerabilidad radica en usuarios utilizando clientes (resolvers) de DNS que no utilizan puertos aleatorios ni identificadores aleatorios para sus consultas, de forma que hacen posible que una atacante realice una predicción de sus próximas peticiones y envíe respuestas falsificadas al usuario. La inmensa mayoría de las peticiones y respuestas de DNS viajan por UDP, y esto hace más sencillo el ataque.

Además de instalar versiones actualizadas de los clientes (y servidores) de DNS, se han definido extensiones de seguridad de DNS. Dichas extensiones permiten la utilización de claves públicas y firma de registros, que se distribuyen por el propio servicio DNS mediante los nuevos tipos de registros de recursos: RRSIG, DNSKEY, DS, NSEC, NSEC3, NSEC3PARAM. Esto es de especial relevancia para los administradores de zonas de Internet, que encuentran un mayor soporte de seguridad a la hora de realizar la transferencia de información de zona entre el servidor primario y los secundarios.

1.9. SERVICIOS PARA ADMINISTRACIÓN DE REDES

Existen diferentes **servicios para administrar y mantener redes**, aunque la mayoría de ellos no son utilizables de forma directa por los usuarios normales, son herramientas para los administradores de red.

Dos **herramientas** comunes de este tipo son **ping y traceroute**. Ambas usan el protocolo ICMP. A diferencia de muchos de los programas que hemos analizado no son clientes de un servidor específico, ICMP se implementa a bajo nivel como parte inseparable de los protocolos TCP/IP.

- **ping** dice si puede o no hacer llegar un paquete a y desde un anfitrión determinado y,

a veces, también suministra información sobre cuánto se tarda en hacer el viaje de ida y vuelta.

- **traceroute** notifica no sólo si se puede llegar a un anfitrión específico (y si puede responder) sino además informa sobre la ruta que siguen los paquetes para llegar a él, lo cual es muy útil para analizar problemas en alguna parte de la red entre el origen y el destino.

No hay riesgos conocidos para ping o traceroute de salida, y muy pocos para los ping y traceroute que entran. Pueden utilizarse para ataques de negación del servicio, aunque la amenaza mayor proviene de que a través de ellos se determine qué anfitriones existen en un sitio como paso preliminar para atacarlo. Por esta razón, muchos sitios evitan o limitan los paquetes que van a entrar. Debido a que no hay servidores específicos para ping y traceroute, para bloquearlos no se puede decidir no encender los servidores, pero se pueden bloquear mediante el filtrado de paquetes.

El **Protocolo Simple de Administración de Redes SNMP** (Simple Network Management Protocol) fue diseñado para facilitar la administración central de equipos de red (enrutadores, puentes, concentradores y, hasta cierto punto, anfitriones). Las estaciones de administración SNMP pueden solicitar información sobre si una interface está activa o no, cuántos bytes se han transferido a través de esa interface, cuántos errores ha habido, etc., también pueden controlar ciertas funciones del equipo de red (activando o desactivando una interface, configurando sus parámetros, etc.). A través de SNMP, se puede obtener información urgente y puntual, como por ejemplo, que una línea no esté operativa o que se están produciendo un número importante de errores en una línea específica. El **riesgo principal de seguridad con SNMP** es que otra persona asuma el control del equipo de red y reconfigurarlo para sus propios propósitos (desactivar el filtrado de paquetes, cambiar el enrutamiento o, simplemente, destruir la configuración).

1.10. SISTEMAS DE FICHEROS EN RED

Existen varios protocolos para montar sistemas de archivos que estén físicamente conectados a otros ordenadores, lo que permite utilizar archivos remotos sin tener que transferirlos de un lado a otro y tratar de mantener múltiples versiones en sincronía (ficheros compartidos). Este servicio es peligroso porque permite leer datos sin obtener una autenticación adicional en la máquina donde éstos residen. Los sistemas más usuales de este tipo son el **Sistema de Archivos de Red NFS** (Network File System) y el **Sistema de compartición de Archivos CIFS**. NFS se diseñó para usarse en redes de área local y brinda respuestas rápidas, gran confiabilidad, sincronización de hora y un alto grado de confianza entre las máquinas. CIFS se diseñó para usarse en redes windows, para abrir la posibilidad de compartir directorios (shares) entre usuarios y máquinas.

Si NFS no se ha configurado de manera adecuada, un **atacante** puede montarlo en el sistema de archivos, y entonces permite, a las máquinas cliente, leer y cambiar archivos almacenados en el servidor sin tener que iniciar una sesión con éste o teclear una contraseña.

A través del archivo **/etc/exports** se permite especificar a NFS qué sistemas de archivos pueden montarse, y qué máquinas pueden hacerlo. Si se deja un sistema de archivos fuera

de /etc/exports, ninguna máquina puede montarlo. Si se pone en **/etc/exports** pero no se especifica qué máquinas pueden montarlo, permite que cualquier máquina lo haga.

Capítulo 2 VIRUS EN INTERNET

Los virus y el malware son el **problema de seguridad más acuciante** en el mundo empresarial. La celeridad con que un virus puede afectar a miles, tal vez millones de usuarios, no era siquiera soñada hace unos años. Gracias al potencial de Internet, un mismo fichero o programa puede llegar a un gran número de usuarios en cuestión de unas pocas horas. Cada día resultan más costosos aunque sólo consideremos el tiempo y los recursos perdidos que provocan.

Otro peligro multiplicado por Internet estriba en la **facilidad de ejecutar programas** sin que el usuario se dé cuenta. Cuando los virus sólo se propagaban por los programas .exe o .com o en las macros de documentos generados por word y excel uno sabía a qué atenerse. Con el advenimiento de contenidos ejecutables en Internet, de la mano de controles Active X, applets de Java, guiones en Visual Basic Script, Java Scripts, plug-in para visualización de películas, o para reproducción de música, se puede llegar a difuminar el concepto de lo que es un programa, un documento o una imagen. En 2017 uno de los ataques más extendidos ha sido la utilización de las máquinas de usuarios que estaban navegando por páginas web para utilizarlas en la minería de criptomonedas. Basta con que la página incluya un código javascript para que el navegador colabore en la minería. En ocasiones el sitio web actuaba de forma consciente para la operación, y en otras era una tercera parte ajena a la web que estaba explotando una vulnerabilidad de XSS y había alojado el código javascript y el identificador del monedero del atacante. Los más perjudicados son los teléfonos móviles, que se pueden calentar en exceso y acabar con su batería en poco tiempo.

Precisamente esas funcionalidades son las que han permitido la **creación de virus de HTML**, es decir, virus cuya infección puede producirse mediante la mera visualización de una página infectada. Este tipo de virus se programa habitualmente en **JavaScript** que se oculta en el código HTML de la página y se ejecutan automáticamente al visualizarse la misma con un navegador. Al ejecutarse, el archivo infecta todas las páginas con las extensiones .htm o .html en el mismo directorio. Los daños varían según el tipo de virus.

Aunque hay otras muchas vías mediante las cuales un ordenador puede llegar a infectarse por medio de la red, el medio estrella de difusión de virus es el **correo electrónico con ficheros adjuntos**. El increíble crecimiento del correo electrónico en los últimos años ha propiciado que los creadores de virus se fijen en este formidable instrumento de comunicación, esto ha convertido a nuestros buzones en focos de peligro potencial.

Algunos programas de correo como el Outlook **son capaces de enviar mensajes** incluyendo botones, formularios y otras formas de automatización. Esta tendencia a la complejidad del correo crecerá en el futuro aumentando las probabilidades de infección mediante el correo electrónico. En estos momentos **debemos tener en cuenta** las siguientes cuestiones:

1. Alguno de los lectores de correo permiten completar el mensaje descargando ficheros y componentes directamente de la web. Esto hace que el mensaje deje de ser un elemento estático, para convertirse en un documento que interactúa con Internet y que puede ser completado con componentes en el momento de ser visualizado.

2. Desde las primeras versiones de Exchange y más tarde con Outlook así como con otros lectores de correo se puede enviar objetos OLE incrustados en los mensajes. No es por lo tanto, un fichero lo que se envía, sino un objeto que es abierto o ejecutado cuando se abre el mensaje.

3. Desde la aparición de Windows 98, que instala por defecto el Windows Scripting Host, que permite lanzar automáticamente programas escritos en JavaScript, Visual Basic script y otros lenguajes, se ha abierto una nueva vía de proliferación de virus. Las características más importantes de estos ficheros son: su facilidad de programación, el total acceso al sistema y a los objetos OLE a través de la automatización. Virus como Melisa utilizan objetos OLE de Outlook y Word que se lanzan automáticamente para infectar y replicarse a través del correo electrónico.

4. Otro aspecto importante es la capacidad de insertar mensajes dentro de otros mensajes (mensajes anidados). Este es un peligro potencial muy grave, pues no todos los antivirus son capaces de analizar este tipo de mensajes.

5. Otra vía de entrada de virus son los formularios y botones en sistemas que permitan insertar scripts en los mensajes como Exchange/Outlook o Lotus Notes.

6. En los últimos tiempos los gusanos han cobrado una fuerza inusitada gracias al correo electrónico. Los gusanos modernos van unidos, en muchas ocasiones, a otras técnicas propias de los virus, y se diferencian de los gusanos tradicionales en que ya no sólo pretenden sobrecargar los recursos del sistema o provocar un colapso en la red, sino que también son capaces de provocar daños en nuestro ordenador o convertirse en troyanos que dejan nuestro equipo en manos de otra persona.

Para **protegernos de los ficheros adjuntos** en el correo electrónico se debe combinar el uso de un buen antivirus, actualizado diariamente, con unas dosis de precaución, para ello se pueden **seguir las siguientes pautas**:

1. No abrir ningún fichero que resulte sospechoso, que venga de un destinatario desconocido o que contenga textos extraños, con cadenas de caracteres sin sentido, o sencillamente que no sean esperados y cuyo contenido no se halle correctamente explicado en el cuerpo del mensaje.

2. En lugar de abrir el fichero, archivarlo y analizarlo con un antivirus actualizado. Los antivirus actuales revisan directamente los ficheros adjuntos incluidos en los correos.

3. Finalmente si el archivo no está infectado pero tenemos reservas sobre su origen, conviene no ejecutarlo. Hay que tener presente que los virus se expanden a una enorme velocidad por

lo que es posible que el virus llegue a nuestro buzón de correo incluso antes de que las empresas de antivirus tengan conocimiento de su existencia

En relación con las **amenazas de los virus** hay **tres medidas básicas** y complementarias: la mentalización y formación de los usuarios, la realización de copias de seguridad y la instalación de un buen antivirus actualizado al día. Estas medidas deben estar encardinadas en la política de seguridad de los sistemas de información de la organización.

Así mismo es muy conveniente **guardar siempre un USB/disco/CD de arranque limpio de virus**. Esto facilitará la limpieza del ordenador en caso de que esté contaminado por un virus.

Para los administradores de sistemas, si están al cargo de las pruebas de penetración como parte de la auditoría de seguridad de la organización, al igual que periódicamente revisan la fortaleza de las contraseñas de los usuarios con herramientas tipo John the Ripper, resulta muy interesante la realización de pequeñas campañas de phishing enfocadas a grupos de usuarios concretos para mostrarles los peligros de abrir correos maliciosos. Por supuesto todas estas acciones deben estar comunicadas y autorizadas.

2.1. ESTRATEGIAS DE PROTECCIÓN ANTIVIRUS EN REDES

El problema fundamental en entornos de red es que **los servidores actúan como trampolines** para que la infección se extienda por todas las estaciones de la red rápidamente. Basta con que haya un programa infectado en un disco compartido del servidor para que cualquier usuario que lo utilice se vea así mismo infectado.

El hecho de que haya un **archivo infectado en el disco del servidor** no significa que el propio servidor esté infectado. Es decir, no significa que el servidor haya ejecutado ese archivo, sino que sólo lo está almacenando.

Generalmente **un servidor dedicado** no se verá afectado directamente por un virus, ya que el virus no se ejecuta en la CPU del servidor. No obstante, existen ciertos riesgos, ya que los virus que están activos en las CPU's de las estaciones tienen los mismos privilegios que los usuarios de esas estaciones. Esto significa que si el puesto del administrador está infectada, nada impide al virus activo en ese puesto borrar lo que le plazca del servidor.

Para proteger un ordenador contra el ataque de los virus es necesario evitar a toda costa ejecutar archivos infectados. La única forma de prevenir estas circunstancias es disponer de un programa residente que vigile constantemente cualquier operación de apertura o ejecución de archivos para detectar la presencia de virus y poder indicar al sistema operativo que cancele la operación en curso.

Al tratarse de redes hablamos de la necesidad de **proteger sus elementos más sensibles**, que dependiendo de los casos pueden ser las estaciones, los servidores o incluso ambos.

Protección únicamente de las estaciones

La instalación en todas las estaciones de la red de un **antivirus residente** protege contra la infección, independientemente de la procedencia de los virus (disquete, red local, Internet, e-

mail, etc.). Como todas las operaciones de apertura de archivos son interceptadas por el programa residente, no importa donde se está intentando ejecutar el programa infectado, ya que siempre será analizado antes de ejecutarse.

Esta **estrategia de protección es básica**, indirectamente se reducen al mínimo los riesgos que pueden acarrear la presencia de un archivo infectado en una unidad de disco del servidor, ya que este archivo será detectado en cuanto un usuario intente usarlo. Además, se evita el problema de que un virus activo en un puesto de trabajo afecte a la información de libre acceso del servidor.

Las **ventajas** de esta solución son:

- No consume recursos del servidor.
- Protege la información de las estaciones de trabajo.
- Reduce las llamadas de soporte.
- Reduce los costes de mantenimiento.
- Evita la proliferación de virus en el parque informático.
- Reduce el riesgo de "exportar" virus al exterior de la compañía.

Los **inconvenientes** de proteger sólo las estaciones son:

- Consume recursos de las estaciones.
- Instalación y mantenimiento complejo.
- No protege directamente los servidores.
- No controla el flujo de archivos entre servidores.

Protección únicamente de los servidores

Esta estrategia implica la instalación **en los servidores de red de un antivirus residente**, que al igual que en las estaciones impida el tráfico de archivos infectados a través del servidor y a su vez impida la infección del propio servidor.

Aunque necesaria, sólo la protección de los servidores es a todas luces incompleta ya que deja a su suerte a las**estaciones de trabajo**, y no hay que olvidar, que hoy en día contienen y procesan cada vez más información.

La protección **únicamente** de los servidores está justificada cuando éstos son la única vía de entrada de información a la red. Por ejemplo cuando todas las estaciones carecen de disqueteras, módems y otros dispositivos de entrada como el correo electrónico.

Ventajas:

- No consume recursos de las estaciones de trabajo.
- Instalación y mantenimiento más sencillos.
- Controla el flujo de archivos entre servidores.
- Protege directamente los servidores.

Inconvenientes:

- Consume recursos del servidor.
- No protege la información de las estaciones de trabajo.
- No impide la proliferación de virus en el parque informático.
- No reduce el riesgo de "exportar virus al exterior de la compañía.

Protección total: las estaciones y los servidores

Es la **solución ideal**, ya que se establecen barreras a todos los niveles, pero a cambio requiere disponer de un conjunto de herramientas que faciliten de forma efectiva la administración de la protección antivirus en todos los elementos de la red: servidores y estaciones, para simplificar al máximo al administrador de la red la gestión de esta protección.

Las **ventajas** de la protección total son la suma de las ventajas de las dos soluciones anteriores menos las que son excluyentes.

Los **inconvenientes** consecuentemente son: Que consume recursos de las estaciones y de los servidores y que la instalación y mantenimiento son más complejos.

2.2. FALSOS VIRUS "*HOAX*"

Desde hace ya unos años vienen difundiéndose por Internet y por todos los servicios de mensajería electrónica una serie de mensajes que hacen alusión a determinados virus que se difunden por e-mail. Según reza la amenaza, al abrirse un e-mail determinado se provocan las más desgraciadas situaciones en el ordenador de la víctima, y recomienda difundir esta advertencia a cuantos más usuarios mejor. La mayoría de estas amenazas son falsas, por lo que se les llama "*hoax*" (engaño).

Este tipo de mensajes se difunde muy rápidamente, a pesar de que en la gran mayoría de los casos no disponen de ninguna base técnica fundada. Todos se basan en el desconocimiento del usuario que recibirá el mail, el cual a su vez remitirá el mensaje a otros tantos usuarios. Así, se

forma una cadena que rápidamente se convierte en miles de mensajes difundiendo el bulo y que consume un gran ancho de banda en los servidores de correo.

Por ello, si recibimos un mensaje de este tipo, no debemos dejarnos llevar por el alarmismo y correr a reenviarlo a todas las direcciones de nuestra agenda. Todo lo contrario, lo más conveniente es olvidarlo y avisar al remitente de la falsedad de su envío. De ese modo se contribuye a que el bulo desaparezca lo antes posible.

2.3. INGENIERIA SOCIAL

A la hora de proteger un sistema informático se disponen **diversos tipos de medidas**: firewalls, VPN, SAIs, backups, discos espejo, cifrado, antivirus... Estas medidas están destinadas a salvaguardar los elementos hardware y software de nuestros sistemas. Sin embargo, la clave fundamental de seguridad es, por encima de todas, el usuario. En ocasiones el desconocimiento, en otras el exceso de confianza, y en la mayoría de las ocasiones una falta de concienciación, pueden hacer vulnerable cualquier sistema independientemente de las herramientas hardware y software con que se cuente.

La ingeniería social es una **técnica muy empleada por los hackers** pero que no hace uso de ningún programa de software, ni elemento hardware, sino que en primera instacia depende tan sólo de la persuasión y psicología del atacante.

Detrás de llamadas telefónicas a un empleado o una secretaria, haciéndose pasar por un técnico en apuros, por otro empleado que necesita ayuda o por un administrador de la red que debe modificar algún aspecto de la configuración, puede **esconderse un hacker**. El atacante aprovechará el exceso de confianza y sus conocimientos de informática para, a través de escogidas y cuidadosas preguntas, ir ahondando en busca de la información que necesita para vulnerar la seguridad del sistema.

La Ingeniería Social no sólo se limita a los ataques de los hackers, sino que suele ser muy empleada a la hora de **introducir los troyanos** e, incluso, en algunos **procesos de los virus** a la hora de infectar.

Hay otras técnicas que han sido utilizadas para introducir **virus en scripts para HTML**. En este tipo de virus se suele provocar un mensaje de aviso por parte del navegador que pregunta al usuario si desea **ejecutar o no el código** que lleva consigo la página web. Una de las últimas técnicas para engañar al usuario y que ejecute el código es superponer una ventana HTML encima del aviso del navegador, con las dimensiones oportunas para que **tape el mensaje** pero deje al descubierto los botones de SI y NO, esta técnica se suele llamar "**espartano**" por su simplicidad. De esta forma la ventana oculta la alarma, y nos presenta un nuevo mensaje de manera que incite al usuario a pulsar el botón SI. A partir de este momento, el virus tendrá control total sobre el sistema y se producirá la infección. Actualmente son frecuentes los ataques registrados por técnicas cross-site scripting, en los que básicamente se utiliza contenido cargado a través de script (o incluso de CSS) de un sitio diferente al de la página web, de forma que el usuario no es consciente y pueden desde falsear el contenido de lo que se presenta (hasta cambiar el DOM), capturar lo que teclea el usuario (keylogger, etc.)

Capítulo 3 PROTOCOLOS DE SEGURIDAD

En las distintas capas del modelo TCP/IP se pueden establecer diferentes controles de seguridad atendiendo al tipo de los datos y a los problemas de seguridad a solucionar, se han diseñado una serie de protocolos en los distintos niveles, que integran algunos de los mecanismos de seguridad ampliamente conocidos y utilizados para solventar dichos problemas.

3.1. REDES PRIVADAS VIRTUALES (VPN)

Un VPN (*Virtual Private Networking*) sirve para transmitir datos de manera segura por una red no segura. Por ejemplo, permite que los usuarios corporativos en viaje puedan conectarse con un Proveedor de Servicios de Internet local ISP y comunicarse de manera segura con su red corporativa.

La principal ventaja de las VPN es la reducción de costes y la mejora de la privacidad. Las compañías pueden reducir sus costes manteniendo una única conexión WAN para cada oficina remota, una conexión a un ISP. El ISP reenvía el tráfico por Internet pública, pero a un coste mucho más reducido.

Las VPN transportan datos a través de un túnel, como se muestra en la figura siguiente. El túnel se crea entre los dos extremos, quienes se ponen de acuerdo en los protocolos de túnel antes de empezar a transmitir datos. Cuando se envían los datos por el túnel la trama o paquete se encapsula dentro de otro paquete. Una vez los datos llegan al extremo opuesto se extraen y se procesan como si se hubieran recibido de la misma LAN.

Básicamente son tres las tecnologías disponibles para crear redes privadas virtuales:

- IPSec (*Internet Protocol Security*), protocolo estándar de seguridad en Internet
- PPTP (*Point to Point Tunneling Protocol*), protocolo de entunelamiento punto a punto
- L2TP *(Layer 2 Tunneling Protocol*), protocolo de entunelamiento de la capa 2

PPTP protocolo de entunelamiento punto a punto

Es una tecnología de entunelamiento multiprotocolo desarrollada por Microsoft para *Windows NT* 4.0. Se basa en el protocolo punto a punto PPP, usado en la mayoría de las conexiones telefónicas y recoge los mismos mecanismos de autenticación y negociación de PPP. Pero a diferencia de este permite crear un enlace virtual que atraviesa redes públicas o privadas.

Aunque sólo pueden actuar como extremo servidor de una conexión PPTP un *Windows NT Server o Windows 2000 Server* cualquier miembro de la familia *Windows* puede ser cliente

L2TP protocolo de entunelamiento de la capa 2

L2TP es una tecnología de entunelamiento multiprotocolo desarrollada, como evolución de PPTP, por Microsoft, Cisco, Ascend, IBM y 3Com. Una de las características más interesantes de L2TP es MPPP, protocolo punto a punto multienlace, que permite conectarse telefónicamente a dos conexiones de ISP distintas. Los datos se transmiten por ambos enlaces a un servidor mediante MPPP de L2TP, donde el servidor ensamblará el tráfico y lo transmitirá a Internet o a una red privada. De esta forma se pueden combinar varios enlaces para obtener un mayor rendimiento .

3.2. PROTOCOLO DE SEGURIDAD DE IP (IPSec)

IPsec permite establecer comunicaciones seguras a través de Internet, independientemente de la aplicación o el protocolo de alto nivel utilizado.

El protocolo utilizado actualmente en la capa IP, el IPv4, originalmente no contempla funciones ni mecanismos de seguridad. Desde 1992, existe un grupo de trabajo en el IETF (*Internet Engineering Task Force*), llamado Ipsec, encargado de la normalización del IPSP (*IP Security Protocol*) y del IKMP (*Internet Key Management Protocol*). Aunque en principio estos trabajos fueron dirigidos para crear la arquitectura de seguridad que debía incorporar el IPv6, también han adaptado esta arquitectura al IPv4.

El IPSP incorpora los siguientes mecanismos:

- La cabecera de autenticación, AH (*Athentication Header*), que permite autenticar el origen de los datos e incluir servicios de integridad.
- Los datos seguros encapsulados, ESP (*Encapsulating Security Payload*), para proporcionar servicio de confidencialidad en los datos.

Ambos mecanismos, AH y ESP, están basados en el concepto de asociación de seguridad, SA (*Security Association*), y pueden ser utilizados conjuntamente o de forma independiente.

Una asociación de seguridad es un convenio entre dos o más partes para decidir sobre los servicios de seguridad que van a utilizar y sobre el proveedor de estos servicios. Los acuerdos a los que llegan las partes implicadas se transmiten como un conjunto de parámetros de seguridad, entre los que se encuentran los siguientes:

- Parámetros de autenticación para la AH (algoritmo, claves, etc..)
- Parámetros de confidencialidad para el ESP (algoritmo, claves, etc...)
- Parámetros del vector de inicialización (IV), o semilla del cifrado simétrico, utilizado para el servicio de confidencialidad (longitud de la clave, utilización o no del servicio, etc...)
- Parámetros de las claves en la SA (periodo de validez).
- Dirección fuente de la SA.
- Nivel de seguridad de los datos protegidos.

Para comprobar que el receptor de un mensaje pertenece a una SA determinada, en caso contrario no podrá autenticar ni descifrar el mensaje, se utiliza una palabra de 32 bits cuyo valor se negocia durante el proceso de gestión de claves. Este valor recibe el nombre de índice de parámetros de seguridad, SPI (*Security Parameters Index*). El SPI, junto con la dirección de destino forman el identificador de una SA.

Las claves de sesión que se pueden utilizar en el protocolo de seguridad IP son de tres tipos:

- Una clave de sesión única entre "*host*"
- Una clave de sesión para usuarios
- Una clave de sesión por aplicación.

Las asociaciones de seguridad necesitan compartir claves que sólo deben conocer los miembros legítimos de una determinada SA. Cuando el número de usuarios es elevado, se necesitan protocolos de gestión de claves eficientes que garanticen la seguridad en la distribución de las claves a todos los usuarios. También ha de tenerse en cuenta la revocación de las claves obsoletas. Con este fin el grupo IPsec ha desarrollado el Protocolo de Gestión de Clave para Internet, IKMP.

3.2.1. CABECERAS DE AUTENTICACIÓN (AH)

Como ya se ha comentado las AH permiten asegurar la autenticidad del origen de los datos, así como la integridad de los paquetes IP durante la transmisión. En algunos casos también está garantizando el no repudio de origen.

Dado que algunos campos en la cabecera del paquete pueden variar durante la transmisión, es necesario calcular los datos que permiten la autenticidad del paquete sobre una copia del mismo en la que los campos susceptibles de modificaciones estén puestos a cero. Sobre esta copia modificada se aplica el algoritmo criptográfico correspondiente (la función MD5, en un principio, pero con tendencia a cambiar a SHA-1 para evitar las colisiones detectadas en la primera).

Aunque los mecanismos para generar las AH pueden implementarse en los "*gateways*", es recomendable hacerlo a nivel de usuario para dar la máxima seguridad.

3.2.2. DATOS SEGUROS ENCAPSULADOS (ESP)

El ESP es un mecanismo para proporcionar confidencialidad y actúa sobre los datos del paquete IP cifrándolos y encapsulándolos. ESP funciona en la capa de red o en la de transporte, por tanto puede cifrar y descifrar los datos de cualquier protocolo de las capas superiores. Cuando se usa ESP en la capa de transporte, se inserta una cabecera entre la cabecera de IP y la de TCP. La información de la cabecera de TCP y todos los datos que contiene el paquete se cifran. Cuando se usa en la capa de red, la dirección exacta de IP de los paquetes se cifra. De esta forma los datos pueden viajar entre dos redes remotas sin que las direcciones IP se revelen a alguien que este analizando el tráfico

En el campo de datos de paquete IP solamente pueden incluirse los datos procedentes de la capa de transporte (funcionamiento en modo transporte) o el paquete IP completo (funcionamiento en modo túnel).

3.3. SSL

El protocolo SSL (*Secure Sockects Layer*, Capa de *Sockets* Segura), introducido por Netscape, trata de construir o ampliar el nivel de *sockets*, añadiendo las características de seguridad necesarias, para que las aplicaciones cliente/servidor puedan comunicarse de forma segura a través de un canal inseguro como es Internet. Cuando el cliente y el servidor se comunican, el socket SSL del lado del cliente y el socket SSL del lado del servidor establecen un canal seguro entre ambos procesos por encima del protocolo de transporte TCP.

De este protocolo existen tres versiones, la última SSL-v3, se puede utilizar para proteger los datos de cualquier aplicación que vaya sobre TCP, ya que, en realidad, lo que hace es introducir una nueva capa, en el modelo Internet, entre los niveles TCP y de Aplicación. Cuando un servidor web está en modo SSL utiliza un puerto distinto (normalmente, el 443) para las comunicaciones cifradas.

El propósito de SSL es ocultar las complejidades de la criptografía a los usuarios y a los desarrolladores. Si los usuarios utilizan un navegador que reconozca SSL, como *Navigator* o *Internet Explorer*, pueden indicarle que cree una conexión cifrada con el servidor con solo reemplazar el "*http*" del URL por "*https*".

Las conexiones seguras SSL proporcionan los servicios básicos de:

- Confidencialidad. Toda la información se transmite cifrada mediante un algoritmo simétrico (DES-CBC, Triple-DES-CBC, RC2 o RC4)
- Autenticación de par. Se puede realizar la autenticación de cada uno de los pares, utilizando cifrado asimétrico (RSA o DSS).
- Integridad. Se incluye un código de autenticación de mensaje (MAC), que permite comprobar el origen y la integridad de los datos (MD5 o SHA-1).

Cuando se inicia la comunicación entre un servidor y un cliente, el primer paso es negociar los algoritmos criptográficos que se van a utilizar. Una vez establecidos los parámetros, todos los intercambios de información entre el cliente y el servidor se realizan cifrados. *Internet Explorer* y *Navigator* muestran un pequeño candado cerrado si la página se descargó con SSL.

Además de estos servicios básicos SSL proporciona otras características interesantes:

- Separación de responsabilidades. SSL utiliza algoritmos independientes para el cifrado, la autenticación y la integridad de datos, con claves diferentes (llamadas *secretos*) para cada función. La primera ventaja de esta separación de responsabilidades es que se pueden utilizar claves mas largas para autenticación e integridad de datos que las que se emplean para confidencialidad, lo cual es útil para los productos a exportar de los EE.UU., ya que los reglamentos federales imponen límites a la longitud de las claves

utilizadas para la confidencialidad pero no a las usadas para garantizar la integridad y la autenticación.

SSL-v3 permite conexiones no cifradas pero si autenticadas y protegidas contra alteraciones deliberadas por un atacante. Esto puede ser útil en circunstancias en las que el cifrado está prohibido por ley, como en el caso de Francia.

- Eficiencia. El cifrado y descifrado de clave pública es una operación costosa en tiempo. En vez de repetir este proceso para cada comunicación entre un cliente y un servidor, las implementaciones de SSL pueden almacenar un secreto maestro, "*master secret*", el cual se conserva entre conexiones SSL Esto permite que las nuevas conexiones SSL inicien la conexión segura de inmediato, sin necesidad de realizar más operaciones de clave pública

- Autenticación en base a certificados. SSL permite la autenticación tanto del cliente como del servidor mediante certificados digitales. SSLv3 utiliza certificados X.509 v3

- Agnóstico en cuanto a protocolos. Aunque SSL se diseñó para correr sobre TCP/IP, puede hacerlo sobre algunos otros protocolos confiable orientado a conexiones, como X.25 u OSI. El protocolo SSL no puede correr sobre un protocolo no confiable, como UDP.

Toda la comunicación SSL tiene lugar a través de un solo flujo bidireccional. En el caso de TCP/IP, los puertos utilizados para los protocolos protegidos por SSL, en general, son:

Palabra clave	Puerto decimal	Propósito
https	443/tcp	HTTP protegido por SSL
ssmtp	465/tcp	SMTP (envío de correo) protegido por SSL
snews	563/tcp	Grupos noticias Usenet protegido por SSL
ssl-ldap	636/tcp	LDAP protegido por SSL
spop3	995/tcp	POP3 (recepción de correo) prtg. por SSL

- Protección de ataques de hombre en el camino y de reproducción. En un ataque de hombre en el camino, el atacante intercepta todas las comunicaciones entre dos partes, haciendo creer a cada una de ellas que se comunica con la otra.

SSL protege contra estos ataques mediante certificados digitales que permiten al usuario del web conocer el nombre válido del sitio web. Desafortunadamente, *Navigator* oculta esta información, siendo accesible solo para los usuarios que seleccionan la opción *Ver* información del documento.

Por ejemplo, en un ataque de reproducción, el atacante puede capturar un mensaje entre un usuario y una entidad financiera, el cual podría, al reproducir el mensaje, indicar la realización

de un pago electrónico varias veces. SSL evita estos ataques al asignar a cada mensaje, dentro de cada sesión, un número de secuencia único.

- Soporte de comprensión. Como los datos cifrados no pueden comprimirse, SSL permite comprimir los datos antes de ser cifrados. Los datos cifrados no pueden comprimirse porque la buena encriptación elimina de forma efectiva todas las repeticiones o similitudes tratadas durante la comprensión.

El protocolo SSLv3, está organizado en dos capas:

- La capa inferior, *SSL Record*, se utiliza para el encapsulamiento y transporte de los protocolos superiores.
- La capa superior, *SSL Message*, compuesta de tres protocolos, que se transmiten sobre la capa anterior:
- Protocolo de *HandShake* de *SSL*, se utiliza para la autenticación entre el cliente y el servidor y para la negociación de los parámetros criptográficos que se utilizarán en la transmisión de los datos.
- *Protocolo de alerta*, son un tipo específico de mensajes que constan de dos partes: un nivel de alerta (*AlertLevel*) y una descripción de la alerta (*AlertDescription*)
- Protocolo *ChangeCipherSpec*, se utiliza para cambiar de un algoritmo de cifrado a otro

3.3.1. SSL HandShake

El protocolo SSL *HandShake* que se ejecuta sobre el protocolo SSL *Record*, permite que cliente y servidor negocien todos los parámetros que necesitan para realizar una transmisión segura.

Cuando un cliente SSL se conecta con un servidor SSL comienza el *handshake* de SSL. El *handshake* de SSL se realiza mediante una conversación de 10 pasos entre el cliente y el servidor. Pero, básicamente realiza lo siguiente:

- El cliente manda un mensaje *Hello* de cliente al servidor; el servidor debe responder con un mensaje *Hello* de servidor, para establecer la conexión. Los mensajes *Hello* se utilizan para establecer las opciones de seguridad que se utilizarán: Versión de protocolo, Identificador de Sesión, Algoritmos criptográficos y Método de Compresión.
- De forma opcional, se pueden intercambiar certificados tanto del servidor como del cliente.

El protocolo SSL *HandShake* se encarga de coordinar los estados del cliente y del servidor en cada sesión y de mantener los estados de sesión y de conexión.

Una sesión SSL puede incluir varias conexiones seguras múltiples. Además, cada una de las partes puede mantener varias sesiones simultáneas. El estado de una sesión incluye la siguiente información:

- Identificador de sesión.
- Certificado del par. Se utiliza un certificado X509.v3.
- Método de compresión.
- Algoritmos de cifrado. Se incluyen el algoritmo de datos (DES) y el algoritmo de MAC (MD5, SHA).
- *Master secret*. Un número de 48 bytes secreto que comparten cliente y servidor.

Por su parte, el estado de una conexión se concreta en:

- Identificación del servidor y del cliente.
- *Server write MAC secret*. Un valor secreto que utiliza el servidor para calcular el MAC en los datos que envía.
- *Client write MAC secret*. Valor secreto que utiliza el cliente para calcular el MAC.
- Clave de escritura del servidor.
- Clave de escritura del cliente.
- Vectores de inicialización. Se utilizan en modos de cifrado encadenados (como CBC).
- Números de secuencia. Cada una de las partes mantiene sus propios números de secuencia de mensajes transmitidos y recibidos.

3.3.2. SSL Record

En la capa inferior del protocolo SSL yace el protocolo SSL *Record*, que se encarga de la transmisión, de forma segura, de los datos que recibe de un protocolo de nivel superior (SSL *HandShake* o cualquier protocolo de aplicación).

La capa de registro de SSL envía bloques de datos, registros, entre el cliente y el servidor. Cada bloque puede contener hasta 16.383 bytes de información. Cada registro SSL contiene la siguiente información:

- Tipo de contenido
- Número de versión del protocolo
- Longitud
- Datos (opcionalmente comprimidos o cifrados)
- Código de autenticación del mensaje MAC

Los aspectos más importantes de este protocolo son:

- Fragmentación. El protocolo se encarga de gestionar la fragmentación y el reensamblaje de los datos que se transmiten, para acoplarse a las características del medio de comunicación.
- Compresión. El protocolo gestiona la compresión y descompresión de los datos, utilizando el algoritmo que se haya negociado.
- Protección con MAC. El protocolo se encarga de construir un MAC para cada mensaje que se transmite, con objeto de verificar la integridad de los datos. Se pueden utilizar distintos algoritmos para construir el valor MAC (MD5, SHA), según la negociación que se haya realizado.
- Cifrado. El cifrado se aplica tras la protección con el MAC, al mensaje inicial más el código MAC; se utilizan distintos algoritmos de cifrado de bloques (DES o RC2), normalmente en el modo CBC.

3.4. TLS

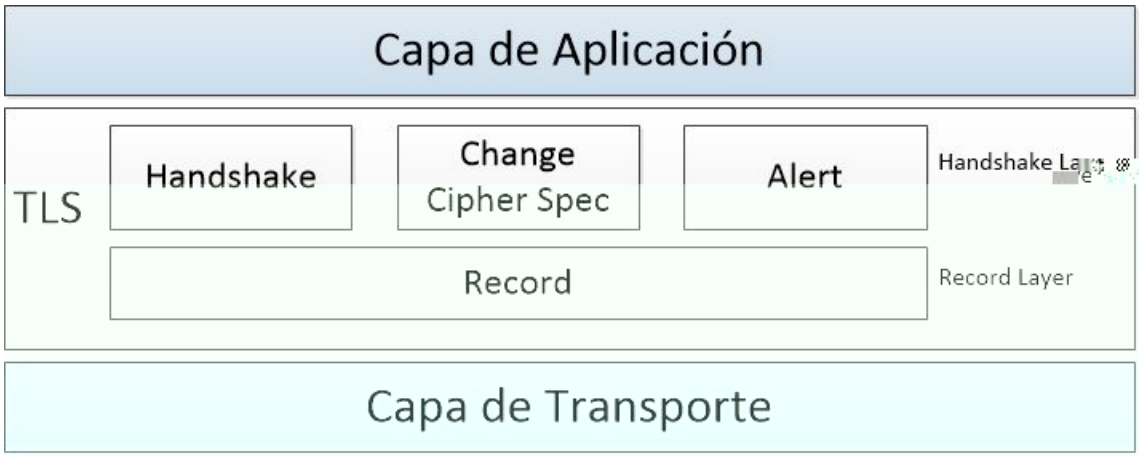
Las tres versiones de SSL se consideran inseguras. Por ello, en enero de 1999, el IETF adoptó una versión mejorada de SSL, que se denominó TLS (Transport Security Layer), y que quedó definida en el RFC-2246. Ésta estaba basada en la versión 3 de SSL (de hecho se suele considerar a TLS1.0 como la versión 3.1 de SSL). Ya en 2008 apareció TLS1.2 (SSL 3.3) que es actualmente la última especificación que existe hasta el momento, definida en la RFC-5246.

SSLv3 y TLS siguen una estructura bastante similar y sólo se diferencian en ciertos detalles, los más importantes se describen a continuación.

- La primera y fundamental diferencia entre SSL y TLS es que TLS es un estándar abierto (no así SSL que fue creado por Netscape). Esto significa que previsiblemente TLS sea adoptado globalmente en mayor medida que otros mecanismos de seguridad o incluso que SSL.
- Otra característica fundamente que diferencia a TLS con respecto a SSL es que en TLS se soportan conexiones tanto seguras como inseguras sobre el mismo puerto, mientras que en SSL si designábamos un puerto para recibir conexiones seguras, este quedaba descartado para recibir conexiones no seguras.

Esto nos permitiría por ejemplo conectarnos a un servidor de correo vía POP o IMAP y poder enviar correos tanto a través de conexiones seguras como inseguras sin tener que cambiar la configuración del cliente de correo para cambiar de puerto. Esto permite que TLS se pueda utilizar en cualquier aplicación, mientras que SSL únicamente se pensó para proteger web y correo.

La siguiente figura muestra la estructura de TLS. TLS está en la capa de transporte.



Como se puede ver se definió para ser implementado en dos niveles: TLS Record protocol y TLS handshake protocol.

- El **TLS Record Protocol** se encarga de transportar los mensajes y aplicar las medidas de seguridad negociadas en el inicio de la conexión con el protocolo de handshake. El protocolo record trocea los datos de aplicación, añade el relleno (padding), cifra y añade el código de autenticación e integridad de acuerdo a las opciones negociadas. Este protocolo puede usarse sin cifrado aunque en caso de necesitar privacidad esta se garantiza mediante el uso de claves simétricas. Se garantiza la seguridad de las conexiones mediante el uso de funciones hash generadas mediante un código de autenticación de mensaje (mejor conocido por si término original, Message Authentication Code o MAC). La forma de calcular este código MAC es una diferencia entre SSL y TLS, ya que en TLS se usan todos los campos (incluyendo el tipo de compresión y la versión) para calcular el código, mientras que en SSL no se utilizaban. En TLS el cálculo de este código MAC puede realizarse con cualquier función de Hash, no sólo MD5 o SHA (que eran los obligatorios en SSL).

- En el **TLS Handshake Protocol** se permiten realizar comunicaciones autenticando tanto al cliente como al servidor. Para ello se realiza la negociación de algoritmos de cifrado y claves antes de que se empiecen a enviar datos por parte de la aplicación. Tres protocolos permiten realizar todas estas tareas:

    - **Protocolo de mutuo acuerdo (handshake). Es el protocolo que permite intercambiar los parámetros de comunicación entre los extremos, incluyendo la versión a utilizar, los argoritmos criptográficos, la autenticación mutua e intercambio de claves.**

    - **Protocolo de cambio de especificaciones criptográficas (change cipher specification)**. Esto permitiría gestionar las transiciones entre las distintas estrategias de cifrado, manejando la nueva negociación de parámetros de

seguridad entre el cliente y el servidor. Este protocolo consta de un único mensaje al final del acuerdo.

- **Protocolo de alerta (alert)**. Define tanto errores que se puedan producir durante la comunicación (que si son fatales pueden provocar la inmediata terminación de la misma), o también mensajes de cierre, para cuando se finaliza la comunicación y evitar así ataques de truncado.

El manejo de la seguridad en el lado cliente es otro aspecto que diferencia a TLS de SSL. Si en SSL, cuando el servidor pedía un certificado al cliente, el cliente le podía contestar con un mensaje de tipo "*no_certificate*". En TLS se ha eliminado ese mensaje, de forma que no es necesario un mensaje distinto en caso de que el cliente no disponga de certificado. Si ocurre esto el cliente simplemente envía un mensaje de certificado vacío. Además, se hace obligatorio el soporte de cifrados basados en Diffie-Hellman, Digital Signature Standard (DSS) y 3DES (en SSL era opcional).

Otro aspecto que diferencia a TLS con respecto a su precedesor es el intercambio de mensajes de verificación de certificados. En SSL esto se hacía mediante el intercambio de varios mensajes siguiendo un mecanismo bastante complejo. En TLS se simplifica este proceso y la información de verificación se obtiene directamente del mensaje de handshake intercambiado al principio de la sesión.

En general el soporte actual de las distintas versiones de TLS es bastante dispar entre unos navegadores y sistemas. La versión TLS1.0 la soportan los principales navegadores actuales (firefox, ,chrome, safari, opera e Internet Explorer). La versión de TLS 1.1 tiene también un soporte bastante extendido, aunque en algunos navegadores venga deshabilitado por defecto (Opera e Internet Explorer). Firefox directamente no lo soporta y tanto Chrome como Safari lo tienen habilitado por defecto. No pasa así con la última versión de TLS, la 1.2. Esta sólo está soportada y habilitada por defecto por Safari. Ni Chrome ni Firefox la soportan y en el casi de Internet Explorer 10 y de Opera 10, está soportado pero no habilitado.

En el caso de los servidores la adopción es bastante dispar. Aunque va evolucionando tanto en las aplicaciones de los equipos de usuarios, fundamentalmente navegadores, y en los servidores.

La siguiente versión de TLS (TLS 1.3) propone varias mejoras respecto a la seguridad y respecto a la eficiencia, aunque hay escepticismo respecto de algunas de estas mejoras. El caso más claro es el 0-RTT propuesto, es decir que el cliente establezca una conexión TLS de forma oportunista y sin necesidad de esperar ninguna respuesta del servidor. Esto solo es posible en algunos casos utilizando información criptográfica conocida previamente.