unir
LA UNIVERSIDAD DE INTERNET

FORMACIÓN PROFESIONAL

UNIDAD FORMATIVA 4

Networking

Seguridad en redes

Índice

Seguridad en redes

Este tema tratará el problema de la seguridad en el acceso remoto de redes. Se trata de un campo extraordinariamente complejo, sobre el que versan innumerables obras, tesis y estudios. Disciplinas enteras de la ingeniería y de las matemáticas estudian las bases teóricas de la seguridad y cómo poder aplicar a nuestro software o implementar en nuestro hardware técnicas que garanticen:

1. La **confidencialidad** de las comunicaciones, de modo que los mensajes intercambiados entre emisor y receptor sean solo visibles por ellos.
2. La **integridad** de los mensajes intercambiados, garantizando que lo que se envía es exactamente lo que se entrega,
3. La **identidad** de las entidades que toman parte, para tener la certeza de que son exactamente quien dicen ser.

Se analizará cómo se utilizan los firewalls (o cortafuegos) para proteger los recursos corporativos de intrusos externos y cómo permitir a las sucursales y a los usuarios remotos acceder a la red interna de forma segura a través de redes públicas no seguras.

El acceso a la red en remoto es una necesidad para la mayoría de las corporaciones. Para la casi totalidad de ellas, internet proporciona una forma económica de conectar sucursales y trabajadores itinerantes a la red corporativa. Esta conexión entre una red corporativa e internet, sin embargo, expone la red interna a riesgos del mundo exterior. Por tanto, las empresas deben tomar medidas para proteger la información confidencial de usuarios externos no autorizados, utilizando la tecnología de red privada virtual o *virtual private network* (VPN).

Hasta ahora, hemos estudiado la seguridad a niveles de acceso a red o incluso más abajo, pero no se ha hablado de la seguridad de las aplicaciones. Vamos a hablar de los principales protocolos de seguridad a este nivel, SSL (*secure sockets layer*) y TLS (*transport layer security),* y de cómo nos ayudarán a enfrentarnos a esos problemas que ya veníamos intuyendo en lecciones anteriores.

Objetivos

- Entender la función de los cortafuegos para controlar el acceso a las redes.
- Saber cómo una VPN permite extender las redes organizativas de forma segura.
- Aprender los fundamentos de la criptografía, necesarios para entender cómo SSL y TLS mantienen la seguridad en la capa 7.

Firewalls

Un cortafuegos (aunque el término cortafuegos es una traducción válida de firewall, el término en inglés está tan extendido que se usarán ambos a lo largo del tema) es un sistema de seguridad que controla el acceso a una red protegida, como una red corporativa, desde otra red. Esta red puede ser una red pública, como internet, u otra red interna. Por ejemplo, es habitual disponer de cortafuegos regulando el tráfico entre la red a la que se conectan los equipos de usuario y las redes con aplicaciones corporativas. Como resultado, cada solicitud de acceso desde la red de origen a la red protegida debe pasar a través del cortafuegos, eliminando la necesidad de protección individual en cada servidor y host de la red protegida.

Para controlar el acceso desde una red pública, el cortafuegos se encuentra en el punto en el que se interconectan las redes. Esta ubicación permite que un mismo dispositivo ejerza de firewall y proporcione autenticación y otros servicios de seguridad a usuarios remotos.

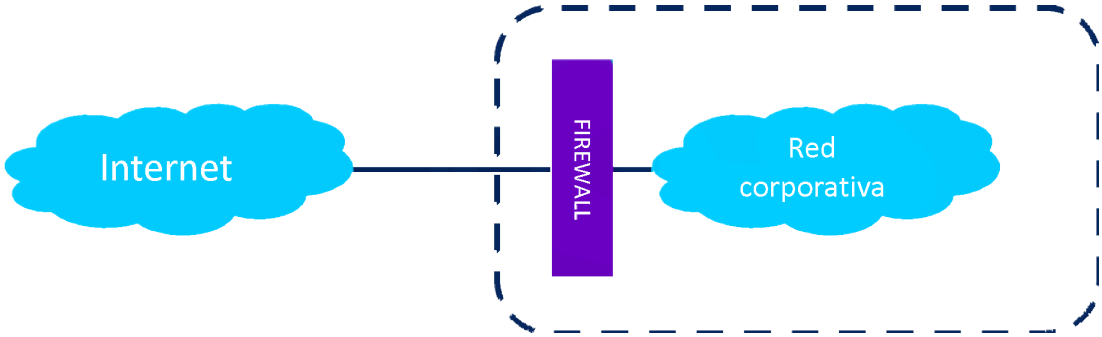

Imagen 1. Diagrama de interconexión entre red pública, red privada y cortafuegos.

Para que un firewall sea efectivo, las organizaciones deben tener bien definida una política de seguridad de red. Esta política identifica los recursos que necesitan protección y las amenazas que existen contra ellos. A partir de estos datos, definen cómo estos recursos pueden ser utilizados y por quién, y establece las acciones a llevar a cabo cuando se violan estas las políticas.

Una política se aplica sobre los dispositivos de red como un conjunto de reglas que comprueban los paquetes que llegan a cada dispositivo. Estas reglas incluyen qué tráfico IP desea permitir la organización para que acceda a su red, qué direcciones de origen deben excluirse de la red, y qué direcciones de destino pueden ser accedidas desde fuera de la red. En cuanto a las acciones específicas, estas incluyen aceptar o rechazar paquetes.

Tipos de firewalls

Los firewalls se pueden clasificar en tres categorías básicas: filtros de paquetes (*packet filters*), filtros de paquetes con estado (*stateful packet filters*) y servidores proxy (que incluyen *gateways* de aplicación y *gateways* de circuito). Hay una cuarta categoría que es esencialmente un híbrido de las tres categorías principales.

Filtros de paquetes

Un filtro de paquetes es un cortafuegos que inspecciona cada paquete de acuerdo con las reglas de filtrado que haya definido el usuario. Por ejemplo, una regla de filtrado podría requerir que todas las solicitudes de Telnet se eliminen. Teniendo en cuenta esta información, el firewall bloqueará todos los paquetes con el puerto TCP 23 como destino (el puerto predeterminado para Telnet). Las reglas de filtrado pueden estar basadas en dirección IP de origen, dirección IP de destino, protocolo de capa 4 (TCP/UDP) y puerto de destino. Por lo tanto, un filtro de paquetes toma decisiones basadas en la capa de red y la capa de transporte. Los filtros de paquetes son rápidos y pueden implementarse fácilmente en routers existentes.

Por desgracia, son los menos seguros de todos los firewalls. Una desventaja que presentan es que no tienen ninguna facilidad de registro que se pueda utilizar para detectar cuando se ha producido una intrusión. Además, un firewall de filtrado de paquetes concede o deniega acceso a la red de acuerdo con las direcciones de origen y de destino y los puertos de origen y de destino. Estos puertos pueden ser falsificados, y como resultado, cualquiera puede acceder a los recursos de la red una vez que se haya dado acceso a un usuario autorizado.

Servidores proxy

Un servicio proxy es una aplicación que redirige las solicitudes de los usuarios hacia servicios basados en la política de seguridad de una organización. Así, un servidor proxy actúa como un intermediario de comunicaciones entre los clientes y los servidores de aplicaciones. Dado que actúa como un punto de control donde se validan aplicaciones específicas, un servidor proxy puede convertirse en un cuello de botella si hay demasiado tráfico.

Los servidores proxy pueden funcionar tanto en la capa de aplicación como en la de transporte. Los primeros se denominan gateway de aplicación y los segundos, gateway de circuito.

Gateway de aplicación

Un gateway de aplicaciones es un servidor proxy que proporciona el control de acceso a la capa de aplicación. Dado que opera en la capa de aplicación, es capaz de examinar el tráfico de la capa más alta en detalle y, por lo tanto, es considerado el tipo de firewall más seguro. Generan registros de todas las actividades y aplicaciones de la red de acuerdo con las necesidades de auditoría de seguridad.

Los gateways de aplicación también pueden ocultar información hacia el exterior. Dado que todos los servicios en la red protegida pasan a través del gateway, este puede proporcionar la funcionalidad de traducción de direcciones de red (u ocultar direcciones IP) y ocultar direcciones IP en la red protegida desde internet, reemplazando la dirección IP de cada paquete saliente (es decir, paquetes que van desde la red protegida a internet) con su propia dirección IP. La traducción de direcciones de red también permite que las direcciones IP no registradas sean libremente utilizadas en la red protegida porque el gateway las mapea a su propia dirección IP.

Gateway de circuito

Un gateway de circuito es un servidor proxy que valida las sesiones TCP y UDP antes de permitir una conexión o un circuito. Está activamente involucrado en el establecimiento de la conexión y no permite que los paquetes se envíen hasta que hayan pasado con éxito las normas de control de acceso. No son tan seguros como los de aplicación porque no analizan la capa superior. Además, una vez que se ha establecido una sesión, cualquier aplicación puede ejecutarse a través de esa conexión. Este comportamiento expone la red protegida a los ataques de intrusos.

Filtros de paquetes con estado

Los gateway de aplicación ofrecen la mejor seguridad, pero tienen también los requisitos de procesamiento más alto, lo que puede reducir el rendimiento de la red. Un filtro de paquetes con estado intenta proporcionar seguridad sin comprometer el rendimiento.

A diferencia de un gateway de aplicación, un filtro con estado comprueba los datos que pasan a través de la capa de red, pero no los procesa. El firewall mantiene la información de estado para cada sesión. Si los paquetes nuevos no pertenecen a una sesión válida, ni intentan crear una sesión que cumple con las políticas del firewall, se rechaza.

Arquitectura del firewall

La arquitectura del firewall se refiere a la forma en que los componentes del cortafuegos están dispuestos para proporcionar una protección eficaz contra usuarios no autorizados. Una red corporativa usualmente tiene múltiples redes perimetrales que se pueden clasificar en tres grupos: la red perimetral más externa, una o más redes internas del perímetro y la red perimetral más interna. El perímetro exterior proporciona un límite entre los recursos corporativos (que necesitan ser protegidos) y los recursos externos (recursos que la corporación no puede controlar). Las redes perimetrales internas representan los límites de los recursos que necesitan seguridad adicional.

Cortafuegos de host dual-homed

Un equipo **dual-homed** es aquel que tiene dos tarjetas de red. Si el host ejecuta un proceso de firewall, las interfaces se pueden aprovechar para una arquitectura de seguridad concreta: una interfaz está conectada a la red interna y otra interfaz está conectada a internet o a alguna otra red no confiable. Por lo tanto, todo el tráfico IP de internet debe pasar por el firewall antes de llegar a un host en la red privada. Del mismo modo, un host interno puede comunicarse con hosts externos a través del host dual-homed.

Cualquier comunicación indirecta que intente esquivar el cortafuegos está bloqueada por diseño, ya que no hay otra conectividad entre las redes más que la que atraviesa el cortafuegos. El host dual-homed puede funcionar como un router para garantizar que internet y la red privada están lógicamente desconectadas, de modo que incluso cuando haya problemas en el sistema, el cortafuegos no falle.

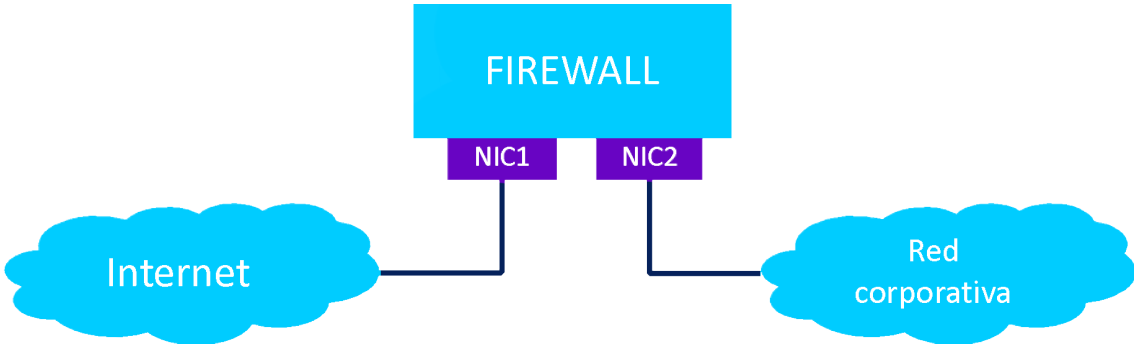

Imagen 2. Firewall dual-homed. Elaboración propia.

Cortafuegos de host de rastreo

En esta arquitectura, el host que actúa como cortafuegos, llamado **bastión**, solo se conecta a la red privada. Un router de rastreo (o de screening) adicional es colocado entre el bastión e internet. Por lo tanto, esta arquitectura combina un enrutador de filtrado de paquetes y un gateway de aplicación.

El router de rastreo realiza una función de filtrado de paquetes y está configurado para que el bastión sea el único host de la red privada al que se puede acceder desde internet. Se puede proporcionar seguridad extra para que el rastreo permita el tráfico solo a puertos específicos del bastión, bloqueando el resto por defecto.

Dado que el anfitrión del bastión es el huésped más expuesto en la red privada, suele ser el más protegido. Generalmente, no hay un único bastión, sino varios. Estos suelen actuar como servidores proxy para servicios públicos como FTP, HTTP o SMTP.

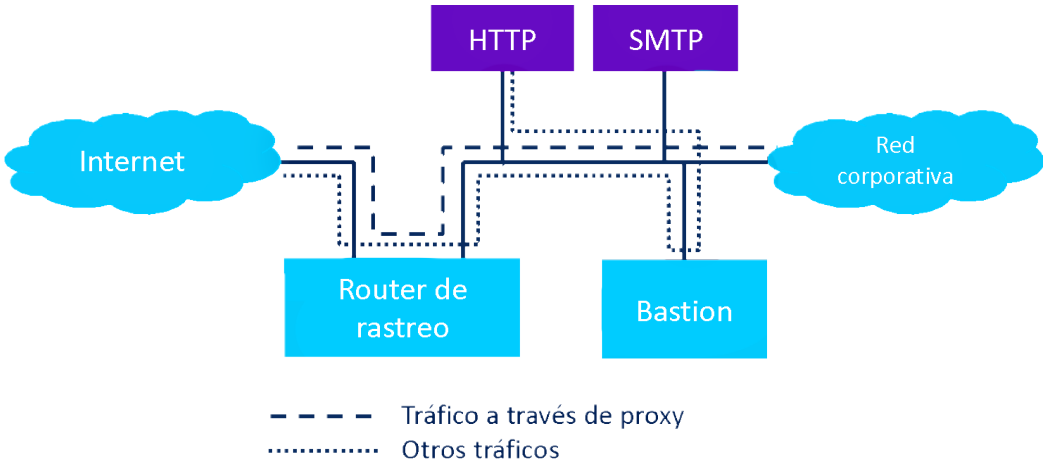

Imagen 3. Firewall con router de rastreo. Elaboración propia.

Cortafuegos de subred o DMZ

El cortafuegos de subred se puede considerar una extensión del cortafuegos de host de rastreo. También incorpora un router de rastreo, denominado externo, y un host bastión. Sin embargo, este cortafuegos crea una capa adicional de seguridad añadiendo una red de perímetro que aísla a la red privada de internet. Esta capa define una DMZ (de *demilitarized zone* o zona desmilitarizada) demarcada por el router externo y un router interno. Este último está localizado más cerca de la red privada que del enrutador externo. El bastión y los servidores de acceso público se encuentran entonces dentro de la DMZ.

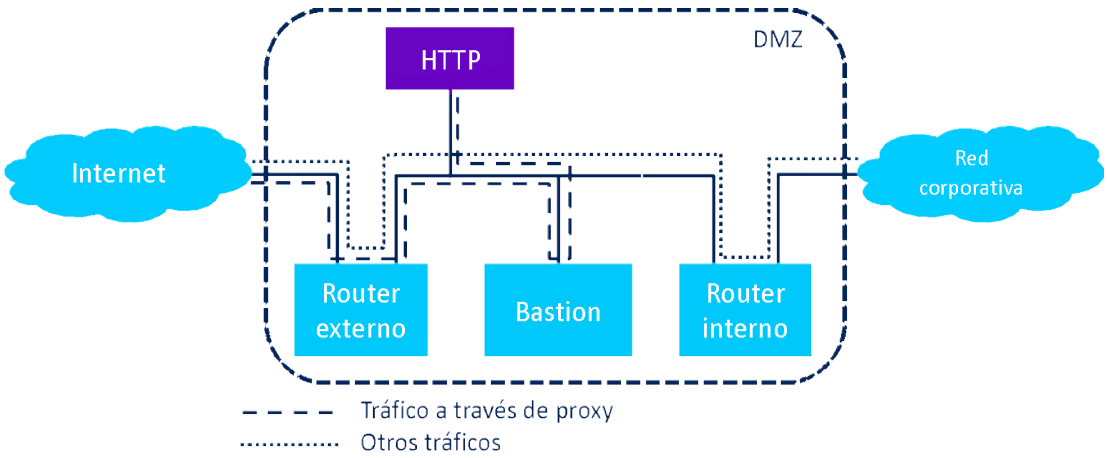

Imagen 4. Arquitectura con DMZ.

La DMZ puede considerarse una red aislada entre el sector privado de red e internet. Para que un ataque llegue a cualquier host interno localizado fuera de DMZ, el atacante debe acceder a ambos routers. Además, su arquitectura revela solo la red DMZ al mundo exterior y mantiene la red privada oculta.

VPN

Una red privada virtual (**VPN**) proporciona una conexión segura entre un origen y un destino a través de una red pública no segura, como internet. Las VPN reducen los costes de acceso remoto. En comparación con otras soluciones, incluidas las redes privadas, una VPN es **barata**.

Una VPN utiliza el cifrado de datos y otros mecanismos de seguridad para que los usuarios no autorizados no puedan acceder a los datos (es decir, leer el contenido de los paquetes) y para garantizar que los datos no se puedan modificar sin detección (es decir, que no alteren los datos en tránsito), a medida que fluyen a través de internet.

Las VPN encapsulan un tráfico en otro, a modo de túnel. En el contexto de internet, el proceso de tunelización permite encapsular protocolos tales como IPX, AppleTalk e IP a través de IP. Efectivamente, IP puede ir encapsulado en IP: el contenido sería el tráfico IP de la red privada y la capa IP externa sería el flujo a través de la red no segura. El proceso de encapsulado cifra el contenido del tráfico interno antes de transportarlo por el túnel y lo descifra una vez en el destino.

Los firewalls y las VPN van de la mano. Mientras que los firewalls controlan el acceso a los recursos de la red corporativa, las VPNs proporcionan privacidad entre dos redes cuando el camino entre ellas no es seguro. Muchos productos de firewall proporcionan acceso seguro desde un cortafuegos hasta una VPN para enviar el tráfico de forma segura.

Ventajas de VPN

Las empresas pueden explotar la naturaleza global de internet y utilizar VPNs para enlazar todas las sucursales en redes privadas. Una organización puede hacer que ciertas secciones de su intranet sean accesibles a sus proveedores estratégicos por medio de la extranet.

Los **túneles VPN** permiten transmitir, a través de subredes no enrutables, segmentos de tráfico específicos de la LAN de la intranet corporativa. Esta característica es útil, especialmente en aplicaciones legacy o heredadas: aplicaciones antiguas, a veces desarrolladas como proyectos de llave en mano, sin soporte y difíciles de mantener.

Tipos de VPN

Se pueden citar tres tipos de VPN en función de la necesidad que cubren dentro de una organización:

- **Las VPN de acceso** proporcionan acceso seguro a las redes corporativas a los usuarios remotos, teletrabajadores y sucursales de oficinas.
- **Las VPN de intranet** permiten que las oficinas remotas y sucursales estén vinculadas a las sedes corporativas de forma segura.
- **Las VPN de extranet** permiten que los clientes, proveedores y socios puedan acceder a la intranet corporativa de una manera segura.

El protocolo PPP

PPP (*point-to-point protocol* o **protocolo de punto a punto**) es un protocolo de la capa de enlace o nivel 2, encargado de establecer una conexión entre dos nodos de una red, usualmente routers, y que proporciona un mecanismo de entramado o *framing* gracias al protocolo **LCP** *(link control protocol)* que delimita cada paquete IP que se envía a través de ese enlace, añadiendo control de errores y posibilidad de autenticación punto a punto usando dos protocolos auxiliares:

- PAP (*password authentication protocol*), en desuso, pues necesita intercambiar una contraseña en claro a través del enlace.
- CHAP (*challenge-handshake authentication protocol)* definido en el RFC 1994 y basado en técnicas criptográficas.

Arquitecturas VPN

Una VPN consiste en los siguientes componentes:

- Un cliente VPN.
- Un servidor de acceso a la red (NAS, de *network access server*).
- Un dispositivo que termina en un túnel (o servidor VPN).
- Un protocolo VPN.

En una conexión tradicional de VPN de acceso, el **cliente VPN** inicia una conexión **punto a punto** PPP con el NAS del ISP a través de la red telefónica. Un NAS es un dispositivo que termina las llamadas de marcación en circuitos analógicos o digitales (RDSI). El NAS es propiedad del ISP y se instala generalmente en el punto del servicio del ISP. Después de que el usuario haya sido autenticado, el NAS dirige el paquete al túnel que conecta tanto al NAS como al servidor VPN. El servidor VPN puede residir en el *point of presence* del ISP o en el sitio corporativo, dependiendo del modelo de VPN que se implemente. El servidor VPN recupera el paquete del túnel, lo desenrolla y lo entrega a la red corporativa.

Protocolos de *tunneling*

Hay cuatro protocolos de **tunelización** utilizados para establecer VPNs. Pueden clasificarse de forma general en dos grupos: PPTP, L2F y L2TP son protocolos de tunelización de capa 2, mientras que IPSec es un protocolo de tunelización de capa 3.

Protocolos de capa 2

Estos protocolos operan en la capa de enlace de datos. Encapsulan paquetes de capa 3 en PPP de capa 2, antes de encapsularlos en IP. Utilizan la seguridad proporcionada por PPP, por lo tanto, realizan la autenticación de usuario con los protocolos de autenticación propios de PPP: PAP y CHAP. No existen disposiciones específicas para el **cifrado** de datos, que puede ser realizado por el usuario antes de encapsular los datos en la VPN.

- PPTP

PPTP, o *point-to-point tunneling protocol*, encapsula los paquetes IPX o IP dentro de paquetes IP. Por tanto, facilita la interconexión de redes no IP a través de internet. Es una extensión de PPP y no soporta conexiones punto a multipunto.

PPTP **no proporciona encriptación paquete a paquete**, delega esa tarea en la PPP. Un paquete PPTP es encapsulado en *generic routing encapsulation* (GRE), el cual es entonces transportado por IP. PPTP separa los canales de control y de datos: el flujo de control corre sobre TCP y el flujo de datos se encapsula sobre GRE (***genering routing encapsulation***), que se transmite directamente sobre IP.

- L2F

L2F es un protocolo propietario desarrollado por Cisco Systems. Soporta una gran variedad de protocolos: puede encapsular tráfico IP, IPX y AppleTalk sobre X.25, IP o ATM. Usa UDP para la encapsulación sobre IP.

A diferencia de PPTP, L2F define su propio encabezado encapsulado, que no es dependiente de IP y GRE. Esta capacidad permite a L2F trabajar en diferentes tipos de redes.

- L2TP

Cuando PPTP y L2F se presentaron a la IETF, la organización decidió combinar las características de ambos protocolos en el protocolo L2TP. A diferencia de PPTP, que se ejecuta a través de TCP, L2TP se ejecuta a través de UDP y no utiliza GRE. Debido a que muchos firewalls no son compatibles con GRE, L2TP es más fácil de integrar que PPTP.

En L2TP, el NAS se llama concentrador de acceso L2TP o LAC, y el servidor VPN se llama servidor de red L2TP o LNS. L2TP utiliza enlaces PPP dial-up y, por ende, también utiliza PAP y CHAP para autenticación, aunque soporta el uso de RADIUS.

L2TP se basa en **IPSec** para realizar el cifrado de datos. Si L2TP descubre que IPSec no es compatible en el extremo remoto, utiliza el cifrado PPP menos seguro. El cifrado puede realizarse desde el puesto de trabajo del usuario o por LAC, dependiendo de la VPN que se utilice.

Protocolos de capa 3: IPSec

IPSec fue diseñado originalmente para incorporar seguridad en la pila TCP/IP. Proporciona **autenticación**, **integridad** y **confidencialidad** a nivel de paquete mediante la adición de dos protocolos: *authentication header* (o AH, encabezado de autenticación), que proporciona integridad de encabezado y autenticación sin confidencialidad; y *encapsulating security payload* (o ESP, contenido con encapsulado de seguridad) que proporciona integridad, autenticación y confidencialidad a la carga útil. Una VPN de IPSec se puede establecer con AH o ESP, o ambos.

ESP permite el encriptado paquete a paquete y utiliza un protocolo de gestión de claves de cifrado basado en estándares. AH no proporciona el cifrado de datos, y es útil en aquellos entornos donde solo se requiere autenticación. También tiene una sobrecarga de procesamiento más baja que ESP.

Un inconveniente de usar IPSec es que solo admite IP, mientras que los protocolos de nivel 2 permiten encapsular más protocolos de nivel 3.

Seguridad a nivel de aplicación

Como hemos visto, las VPNs permiten asegurar el tráfico de capas 2 y 3, pero no hemos hablado de cómo proteger a las aplicaciones que residen en la capa de seguridad: de esto se encargan los protocolos SSL y TLS.

Antes de hablar de los protocolos en sí, se presentarán conceptos de criptografía necesarios para entender por qué son seguros y confiables: algoritmos, funciones de resumen, firmas, certificados y autoridades de certificación.

Los algoritmos criptográficos y las funciones resumen requieren de una base matemática muy profunda, más allá del alcance de esta asignatura. En la sección 'A fondo' se menciona un libro sobre estos temas con un enfoque muy práctico.

Algoritmos

En una situación de ejemplo, un usuario A quiere enviar un mensaje confidencial a un usuario B. El mensaje contiene información personal, por lo que el usuario A querría que sea privado y solo legible por el usuario B. Una solicitud de transferencia a un banco es un claro ejemplo, ya que el mensaje incluye datos personales del usuario A.

Un algoritmo criptográfico permitirá al usuario A convertir el mensaje inicial en un mensaje **cifrado,** ilegible a menos que se le aplique la técnica complementaria de **descifrado**. Para el cifrado se usa una clave secreta que solo deben conocer los usuarios A y B, ya que esa clave se usará en el proceso de descifrado. Se podría intentar descifrar el mensaje por fuerza bruta, pero el objetivo del algoritmo es que no merezca la pena: **los buenos algoritmos criptográficos hacen que sea tan difícil para los intrusos decodificar el texto original que no vale la pena su esfuerzo.**

No siempre en todos los casos la clave que cifra el mensaje original es la misma que descifra el mensaje. Esto depende del tipo de algoritmo.

Criptografía simétrica

Esta familia de técnicas, también conocida como **criptografía convencional**, usa la misma clave para cifrar y para descifrar los mensajes. Por tanto, requiere que ambos extremos de la comunicación (es decir, los usuarios o los agentes de software) compartan la clave.

Estas técnicas tienen un punto débil en el intercambio de la clave: necesitan un canal seguro para compartirla. Si se ha podido compartir en secreto (por un segundo canal confiable o en persona), el algoritmo puede ser tan seguro como permita técnicamente. Es decir, el problema del intercambio de la clave no es una propiedad de seguridad de cada algoritmo simétrico, sino un problema de esta familia en general. La siguiente familia de algoritmos viene a dar una solución al problema.

Criptografía asimétrica

Estos algoritmos también se conocen como **de clave pública**. Resuelven el problema del intercambio de claves definiendo un algoritmo que usa dos claves (conocidas como *key pair* o pareja de claves), cada una de las cuales puede usarse para cifrar un mensaje. Cuando se cifra un mensaje con una de las claves hay que usar la otra para descifrarlo.

Los usuarios ya no tienen que compartir una misma clave, sino que es suficiente que compartan una de las dos claves. Siguiendo con el ejemplo anterior, si el usuario A quiere que solo el usuario B lea el contenido del mensaje, cifrará el mensaje con la clave pública de B. Es decir, B siempre compartirá la misma clave, considerada pública, con cualquier usuario de quien necesite recibir mensajes. **Deberá mantener la clave privada en secreto**. Esta clave privada será la única que permite descifrar los mensajes cifrados con la clave pública de B. No hace falta compartir la clave pública por un canal seguro, hay otras implicaciones a la hora de compartir esta clave que se detallan en los siguientes apartados.

Funciones de resumen (*hash functions*)

Hasta ahora se ha hablado del problema de la confidencialidad, pero no de la **integridad**. Si un usuario C interceptase el mensaje de A, C puede modificar el mensaje a su gusto, o sustituirlo completamente, sin que B tenga constancia de si el mensaje es el mismo que ha enviado A. La integridad se puede conseguir si el usuario A genera un resumen (*hash)* a partir del mensaje original y lo envía a B.

A la recepción de ambos, B calcula el resumen con el mismo algoritmo que usó A y compara el resumen recibido con el generado. Si los resúmenes coinciden, B puede estar seguro de que el mensaje no ha sido alterado en tránsito.

Estos resúmenes se calculan con unas funciones matemáticas llamadas funciones de *hash* o *digests*. Las funciones *hash* reciben un mensaje de longitud variable y generan un resumen de longitud fija, mucho menor. Entre las propiedades deseables de estas funciones están:

- Que sean rápidas.
- Que sea difícil obtener el mensaje a partir del resumen.
- Que la probabilidad de que dos mensajes diferentes tengan el mismo *hash* sea despreciable.

Estas propiedades aumentan la dificultad de sustituir un mensaje por otro para un mismo resumen dado.

El problema de la integridad del mensaje que solucionan los resúmenes queda eclipsado con el mismo problema que había al enviar el mensaje original: si el resumen no se transmite de forma segura no sirve de nada. Si el usuario C intercepta tanto el mensaje como el resumen, nada le impide crear su propio mensaje y generar un resumen acorde al mismo. La solución a este problema son las firmas digitales.

Firmas digitales y certificados

Firmas digitales

En el escenario de ejemplo, el interés de ambas partes es que el mensaje sea secreto y que el mensaje que llega a B proceda de A sin que C haya alterado su integridad. Una solución pasa por generar el resumen a partir del mensaje y **firmar el resumen** con la clave privada de A (ojo, no con la clave pública). B sigue estos pasos cuando recibe la transmisión:

- Descifra el mensaje usando su clave privada. Con esto, han asegurado la privacidad del mensaje.
- Descifra el resumen usando la clave pública de A. Con esto, B se asegura de la autoría del mensaje.
- Calcula el resumen del mensaje localmente y lo compara con el resumen descifrado. Si coinciden, la integridad del mensaje está también asegurada.

Certificados

El proceso anterior es teóricamente válido salvo por un aspecto: con la información que tienen, ninguno de los dos puede asegurar que la clave pública del otro es de realmente quién dicen ser. En estos casos, hace falta una tercera parte en la que ambos confíen.

Esta tercera parte se denomina autoridad de certificación (CA, de *certificate authority*). Las CAs emiten certificados que contienen la clave pública del sujeto en cuestión y están firmados por la propia CA. Así, si tanto A como B confían en la CA, pueden confiar en que la clave pública del certificado de A es realmente de A, y lo mismo ocurre con el certificado de B. Más detalles sobre certificados se añaden en la sección correspondiente en el Anexo I: Contenido de un certificado.

SSL y TLS

SSL es un protocolo de aplicación que actúa como capa intermedia entre TCP y otro protocolo de aplicación. El ejemplo más conocido es HTTPS, en el que HTTP se transmite sobre SSL y usa el puerto 443 por defecto, pero se usa con otras aplicaciones conocidas como FTP y SMTP.

Ofrece un canal seguro que soporta autenticación, integridad y confidencialidad mediante certificados, firmas y cifrado. El protocolo usa criptografía asimétrica para la autenticación y el establecimiento de parámetros de la sesión y criptografía simétrica para el intercambio de los datos.

El protocolo es versátil en cuanto a la variedad de algoritmos que soporta. Ambos extremos de la comunicación deben ponerse de acuerdo en un conjunto de algoritmos de firma y cifrado durante el establecimiento. Esto permite que una organización restrinja el uso de algoritmos menos seguros mediante opciones de configuración, sin necesidad de cambiar la capa de software completamente. Esto hace, además, que el protocolo sea extensible. Las nuevas versiones soportan algoritmos nuevos a medida que están disponibles.

El protocolo que se emplea actualmente es TLS. Hubo varias versiones de SSL con importantes errores de seguridad, que se trataron de corregir con SSLv3 en 1996. A pesar de eso, y tras la última vulnerabilidad detectada en su sistema de cifrado de bloques en cadena (CBC), desde junio de 2015 SSLv3 se considera oficialmente en desuso (*deprecated*).

TLS 1.0 se definió en enero de 1999 como una actualización de SSLv3, aunque con el paso del tiempo la divergencia fue suficiente como para que se consideren protocolos separados. Estudiaremos los fundamentos comunes de ambos protocolos en esta sección, para mayor detalle se recomienda consultar el RFC correspondiente.

Establecer una sesión

El establecimiento de un canal SSL se lleva a cabo con una negociación en la que cliente y servidor acuerdan los algoritmos que van a usar e intercambian los certificados. Este paso es opcional en ambos extremos; por ejemplo, los servidores web envían su certificado durante una petición HTTPS, pero no tienen por qué solicitar un certificado de cliente.

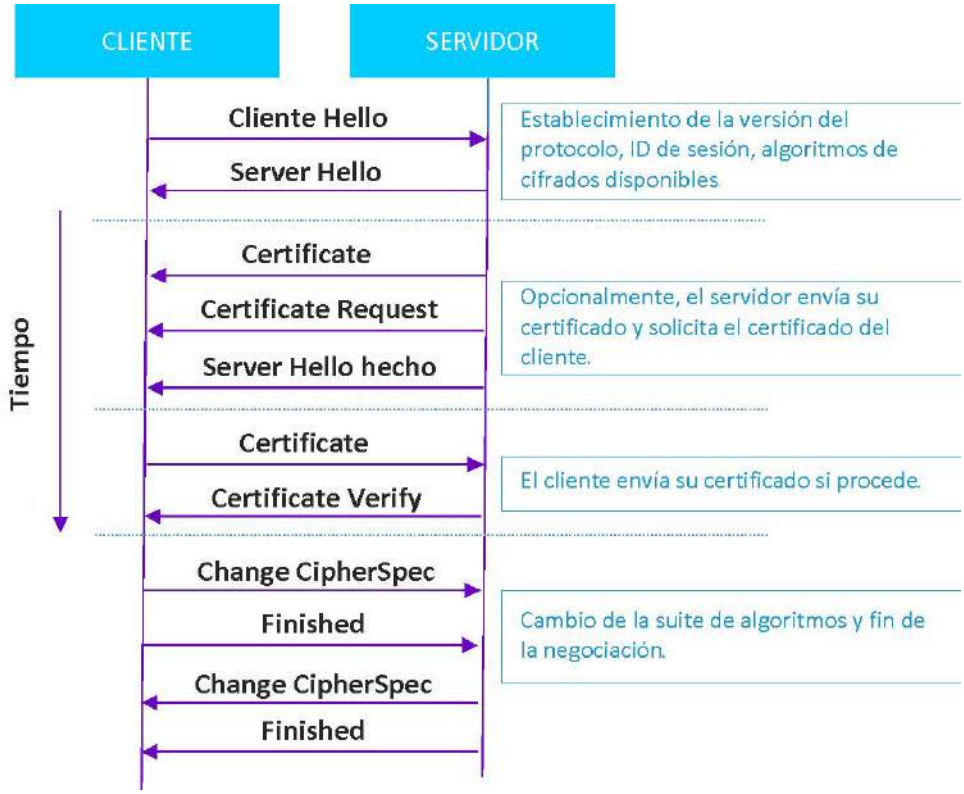

Imagen 5. Establecimiento de conexión SSL.

Esta negociación establece una sesión que se puede reaprovechar para intercambiar tráfico adicional después del intercambio inicial. Esto reduce la latencia inicial para el envío de nuevas peticiones.

La negociación como tal, así como la fase de intercambio de suites, se llevan a cabo con protocolos específicos, todos ellos incluidos en el SSL *record protocol*.

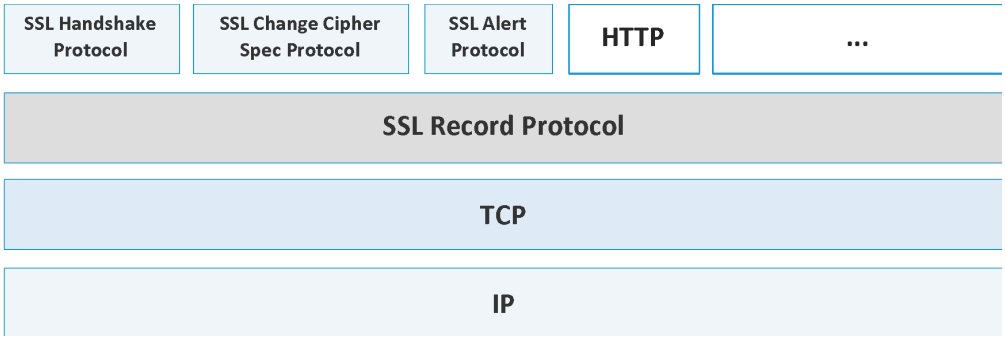

Imagen 6. Pila de protocolos SSL.

La negociación incluye los siguientes pasos:

- Negociación del conjunto de algoritmos.
- Establecimiento de una clave de sesión (esta clave se usará para el cifrado simétrico durante la sesión y no es la clave privada ni pública de ninguno de los extremos).
- El cliente valida el certificado del servidor, si procede, y viceversa.

Durante el primer paso, cliente y servidor acuerdan qué algoritmos usarán en función de los que tengan disponibles: un cliente puede no disponer de los algoritmos más nuevos y un servidor puede no permitir el uso de algoritmos menos seguros, como se ha mencionado anteriormente. Los algoritmos se acuerdan en suites, que son conjuntos de algoritmos que abarcan el intercambio de claves, el algoritmo de cifrado de mensajes y la función de firma.

Método de intercambio de claves

Los datos se intercambian con un cifrado simétrico para reducir el coste computacional del cifrado. Este tipo de cifrado requiere una clave compartida por ambos extremos y para asegurar la privacidad, esta clave debe intercambiarse de manera segura. Algunas técnicas de intercambio de esta clave son RSA y Diffie-Hellman.

Cifrado para transferencia de datos

Este componente de la suite de cifrado se refiere al algoritmo simétrico que se usará durante el intercambio de datos (este algoritmo será el que use la clave intercambiada con el método anterior).

Las versiones anteriores a TLS1.3 ofrecían la opción de no cifrar el contenido (por lo que ofrecían autenticación e integridad, pero no privacidad). El cifrado puede conseguirse con cifrados de flujo (stream cipher) o con cifrados de bloque. TLS 1.2 soporta, entre otros, RC4 como cifrado de flujo y algunas variantes de AES como cifrado de bloque.

Función de resumen o *hash*

TLS 1.2 admite, entre otras funciones, MD5 (un *hash* de 128 bits), SHA-1 (de 160 bits) y variantes de SHA-2 (de hasta 512 bits). Estas funciones generan un código de autenticación de mensaje (MAC, *message authentication code*) a partir del mensaje original. Este código se cifra junto al mensaje como comprobación de integridad.

Transferencia de datos

El SSL *record protocol* encapsula tanto los datos de clientes como la información de control. Realiza las funciones, entre las que se encuentran:

- Fragmenta estos datos en unidades más pequeñas si es necesario y combina múltiples mensajes de datos de protocolo de nivel superior en paquetes individuales.
- Genera el código de autenticación aplicando la función de firma al mensaje.
- Cifra cada fragmento y su firma.
- Delega la transmisión a la capa TCP.

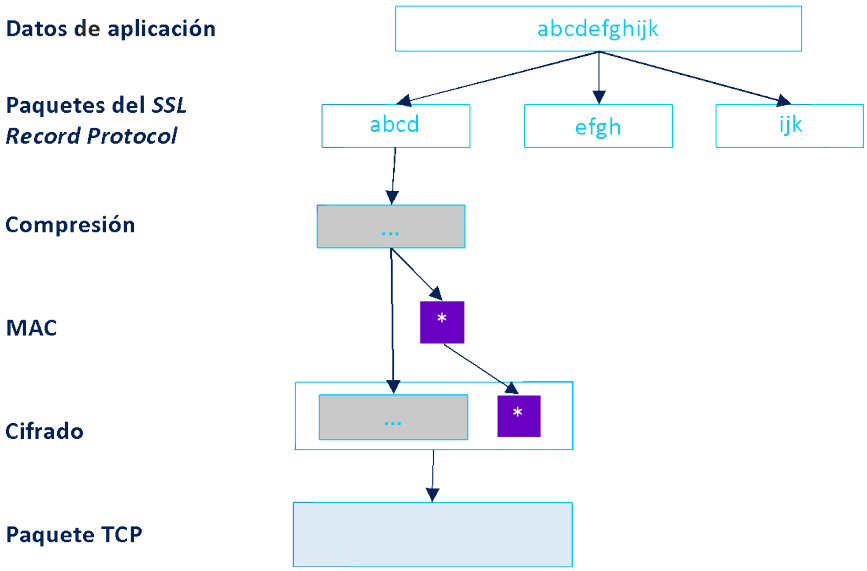

Imagen 7. SSL *record protocol*.

Anexo I: Contenido de un certificado

Al hablar de los certificados, en la sección correspondiente a la seguridad de aplicación, se expusieron las CAs (autoridades de certificación) como el agente necesario para poder establecer una relación de confianza entre los agentes que toman parte en un intercambio digital de información.

Esta confianza en una tercera parte permite validar la identidad real asociada a un certificado. Esta identidad puede ser una persona física o jurídica, un servidor (por ejemplo, identificado por el dominio) u otro tipo de entidad. La tabla 1 resume el tipo de información que incluye un certificado y que identifica al sujeto. Aparte de los datos identificativos, los certificados incluyen otros metadatos, como el periodo de validez o identificadores para uso de la CA.

Información contenida en un certificado	
Sujeto	Nombre (*distinguished name*), clave pública.
Emisor	Nombre (*distinguished name*), firma.
Periodo de validez	Fecha de inicio y fecha de fin de validez.
Información administrativa	Versión, número de serie.
Información extendida	Uso del certificado, información de revocación, etc.

Tabla 1. Información contenida en un certificado.

El sujeto se identifica con un *distinguished name*, o nombre distinguido, en vez de un nombre común. Por ejemplo, el usuario A puede disponer de un certificado digital firmado por la policía del país en el que el nombre distinguido contenga su número de DNI, ya que, en el contexto de los individuos del país, el DNI lo identifica unívocamente.

Por otro lado, el mismo usuario A puede recibir un segundo certificado digital por parte de su empresa en la que el nombre distinguido sea el número de empleado seguido de su departamento y los datos de la empresa. Ambos certificados identifican al mismo individuo, pero cada uno en su contexto. Se definen mediante el estándar X.509 y siguen una estructura a base de componentes y siglas que pueden consultarse en la Tabla 2.

	Información de nombre distinguido		
	Abr.	Descripción	Ejemplo
Nombre común	CN	Nombre que identifica el certificado.	CN=John Doe
Organización, compañía	O	El CN está asociado con esta organización.	O=UNIR
Unidad organizativa	OU	El CN está asociado con esta OU (departamento, sección, etc.).	OU=Facultad de Historia
Ciudad, localidad	L	El CN está ubicado en esta ciudad.	L=Madrid
Estado, provincia	ST	El CN está ubicado en este estado.	ST=Madrid
País	C	El CN está ubicado en este país.	C=ES

Tabla 2. Campos del nombre distinguido.

El formato de los nombres distinguidos es versátil y cada CA puede especificar su propia estructura, definiendo qué campos son requeridos. Por ejemplo, los navegadores web requieren que el CN de un certificado que representa un servidor coincida con un patrón comodín para el nombre de dominio de ese servidor, como *.unir.net. La imagen 8 muestra un certificado de este tipo en el que el *common name* coincide con este dominio.

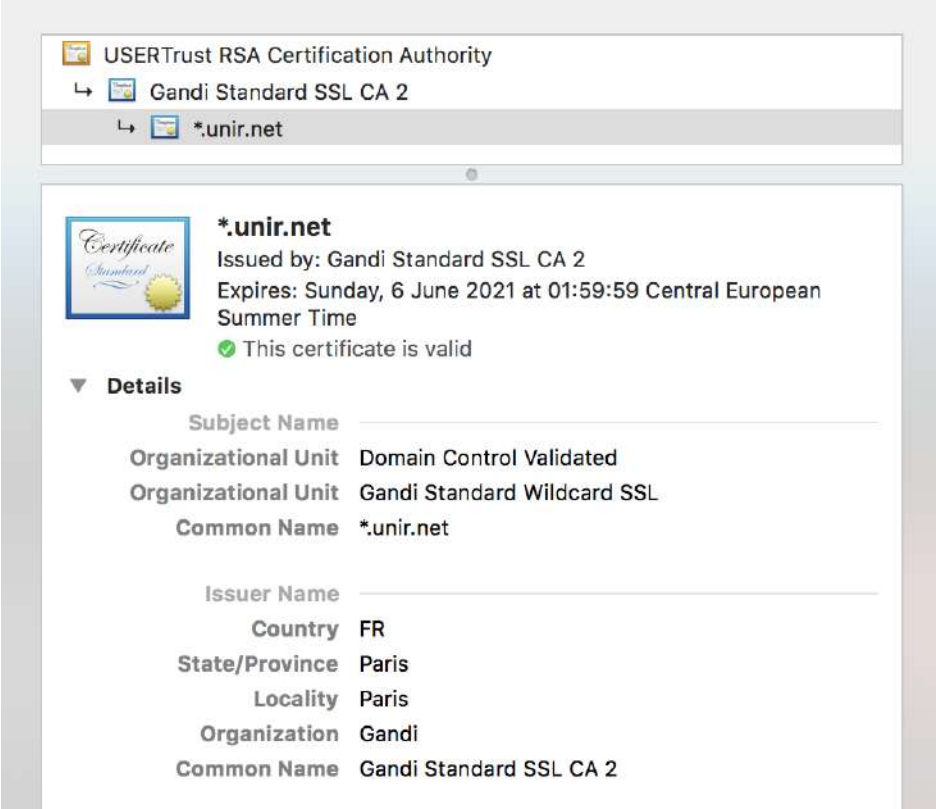

Imagen 8. Certificado de *.unir.net que muestra el CN.

Los certificados siguen el formato binario ASN.1, aunque cuando la transmisión no puede ser binaria se traducen a formato ASCII utilizando la codificación Base64. Este contenido se encapsula entonces en un archivo PEM (*privacy enhanced mail*), en el que se delimita el contenido en Base64 entre dos líneas, tal como se muestra a continuación.

-----BEGIN CERTIFICATE-----
MIICkzCCAfwCCQCNtXKhyVwgRzANBgkqhkiG9w0BAQsFADCBjTELMAkGA1UEBhMC
RVMxDzANBgNVBAgMBk1hZHJpZDEPMA0GA1UEBwwGTWFkcmlkMQ0wCwYDVQQKDARV
TklSMR0wGwYDVQQLDBRGYWN1bHRhZCBkZSBIaXN0b3JpYTERMA8GA1UEAwwISm9o
biBEb2UxGzAZBgkqhkiG9w0BCQEWDGRvZUB1bmlyLm5ldDAeFw0yMDA1MTAxNDU3
NDZaFw0zMDA1MDgxNDU3NDZaMIGNMQswCQYDVQQGEwJFUzEPMA0GA1UECAwGTWFk
cmlkMQ8wDQYDVQQHDAZNYWRyaWQxDTALBgNVBAoMBFVOSVIxHTAbBgNVBAsMFEZh
Y3VsdGFkIGRlIEhpc3RvcmlhMREwDwYDVQQDDAhKb2huIERvZTEbMBkGCSqGSIb3
DQEJARYMZG9lQHVuaXIubmV0MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCl
Tjo92ngKZmfTA3oiJh8VDr13Nz66w7wH1Fv4KV/aDCQJwJ3Wtmfb+lEFsDgV48qE
c+j5ck3ZEKoOkcUu7iJKHgY57tPJLhaUCeJs4C6RYKdLG31A0XEYxj6fqHpsOHhC
lXSfPZpb2plnMeNVdreTfWoMlCxXbk9ueGqjR0NgtQIDAQABMA0GCSqGSIb3DQEB
CwUAA4GBAAX5j+cImEDjWYqBQUZYtYZHt6Ur+DdA3tAHh4F0HOTLxCxPjJl/9bAJ
RS8TaUJ2vdex+Rj43wFch/lMuay1WzbuhRQPEXfzVW5ZtUMdxyFY//JjrDZ8M7FU
pK93iw3r7c9sH8Ql+IPNVj588FVwlCKk1HS2nbsuxyTiRzzRNWCh
-----END CERTIFICATE-----

Autoridades de certificación

Las CA se encargan de verificar la información en una solicitud de certificado antes de concederlo. Es habitual que la CA solo se asegure de que el individuo posee las claves, no de generarlas. Por ejemplo, un método para verificar los certificados para dominios de internet es que el propietario del dominio incluya un contenido firmado en un registro DNS de tipo TXT.

Si la autoridad de certificación puede leer el registro DNS y descifrarlo con la clave pública del solicitante, podrá emitir un certificado porque ha verificado que la clave que ha usado es del propietario del dominio, ya que este ha sido capaz de modificar los registros DNS de ese dominio. Otras formas de verificación pueden incluir una visita presencial, por ejemplo, a una comisaría para conseguir un DNI electrónico.

Cadenas de certificados

Las CAs pueden emitir certificados para otras CAs. Por ejemplo, en los certificados de dominio se permite que haya muchas más autoridades de segundo nivel que de primero. Las de primer nivel emiten certificados de CA y las de segundo, certificados de usuario. Así se consigue, por ejemplo, que los usuarios tengan más oferta y no dependan de un número limitado de CAs de nivel raíz. La imagen 8 muestra la cadena de certificados del dominio unir.net tal como aparece en un navegador web.

CA de nivel raíz

Se han mencionado algunos métodos para verificar la identidad del sujeto antes de emitir un certificado (modificación de registros DNS y visita presencial). También se ha hablado de que los certificados pueden formar una cadena hasta la CA de nivel raíz. ¿Cómo se garantiza entonces que el certificado de la CA de nivel raíz es realmente de la CA que dice ser? Los certificados de estas CAs están firmados por la propia clave privada de la CA, ya que no hay un certificado de una capa superior que lo pueda firmar.

En el caso de los certificados de dominio, los navegadores distribuyen los certificados de las CAs de nivel raíz más extendidas. En una organización con una CA interna, una posibilidad es distribuir los certificados a través de las directivas de grupo de Active Directory.

Varias empresas, como Thawte, VeriSign o Let's Encrypt, se han establecido como autoridades de certificación raíz: se encargan de verificar las solicitudes de certificado (con el método de los registros DNS, por ejemplo), procesarlas, emitir los certificados y gestionarlos.

Gestión de certificados

La tarea de emisión de certificados no es la única que llevan a cabo las autoridades de certificación. Los certificados tienen un ciclo de vida, y uno de los momentos de este ciclo es la posible invalidación de estos. La fecha de expiración invalida un certificado de manera automática, de manera que un cliente puede verificar la validez temporal simplemente leyendo los metadatos del certificado (que, al estar firmados, se tiene garantía de que son válidos) y teniendo el reloj local en hora.

Sin embargo, un certificado puede ser invalidado antes de su fecha de expiración por parte de una CA por varias razones. Por ejemplo, se puede considerar que el algoritmo con el que se firmó ya no es lo suficientemente seguro o que el tamaño de la clave es muy pequeño.

También puede ocurrir que un individuo deje de cumplir los requisitos que le permitieron conseguir el certificado y, si este sigue en vigor, le otorgarían permisos que ya no debe tener.

En ambos casos, una CA puede proactivamente invalidar un certificado. Dado que los certificados son autocontenidos y se pueden distribuir libremente, la invalidación no puede perseguir el borrado ni el envío de un contra-certificado a todos aquellos usuarios que recibieron el mensaje original, por motivos de escala. El método que siguen las CAs es publicar listas de certificados que han sido revocados. Estas listas se conocen como listas de revocación de certificados (CRL, *certificate revocation list*).

En el escenario que se ha usado como ejemplo en este capítulo, B tiene una tarea más para comprobar que el mensaje viene realmente de A: además de comprobar la firma, el resumen y que el certificado ha sido emitido por una CA de confianza, también debe consultar la CRL de dicha CA. Si el certificado de A se encuentra en esta lista, B no deberá confiar en el mensaje.

Bibliografía

- TLS v1.3 - RFC 8446 - TLS v1.3.

LA UNIVERSIDAD EN INTERNET | FORMACIÓN PROFESIONAL

PROEDUCA