

1- INTRODUCCIÓN A LA SEGURIDAD	5
1.1 ANTECEDENTES	6
<i>Introducción a la Sección 1.1</i>	6
<i>Antecedentes</i>	6
1.2 DEFINICIÓN Y OBJETIVOS DE LA SEGURIDAD	7
<i>Introducción a la Sección 1.2</i>	7
<i>Definición</i>	7
<i>Aspectos generales</i>	8
<i>Objetivo de la seguridad</i>	10
<i>Resumen</i>	11
2- ASPECTOS DE LA SEGURIDAD.....	12
2.1 VULNERABILIDADES Y AMENAZAS	13
<i>Introducción a la Sección 2.1</i>	13
<i>Clases de vulnerabilidades</i>	13
<i>Tipos de amenazas por su origen</i>	15
<i>Tipos de amenazas en redes</i>	16
<i>Tipos de amenazas por su naturaleza</i>	20

Amenazas pasivas.....	21
Amenazas activas.....	22
2.2 MEDIDAS DE SEGURIDAD.....	24
Introducción a la Sección 2.2.....	24
Definición y tipos de contramedidas	24
Medidas físicas.....	25
Medidas lógicas.....	26
Medidas administrativas	28
Medidas legales.....	29
Plan de Contingencia.....	31
Principios fundamentales de la Seguridad.....	32
Resumen	34
3- SEGURIDAD EN REDES IP	35
3.1 SEGURIDAD EN REDES IP	36
Introducción a la Sección 3.1.....	36
Definición.....	36
Servicios de seguridad en redes IP.....	37
Mecanismos de seguridad	39
3.2 AMENAZAS A LA SEGURIDAD EN REDES IP	41
Introducción a la Sección 3.2.....	41
Introducción	41
Spoofing o suplantación	42
Hijacking o secuestro de sesión.....	43
Electronic Eavesdropping (sniffing).....	44
The man in the middle	45
Resumen	46
4- CRIPTOGRAFÍA	47
4.1 INTRODUCCIÓN.....	48
Introducción a la Sección 4.1	48
Definición.....	48
Algoritmos y claves de cifrado	49
Ventajas de los algoritmos de cifrado con claves.....	50
La longitud de las claves.....	51
4.2 CRIPTOGRAFÍA SIMÉTRICA.....	52
Introducción a la Sección 4.2.....	52
Definición.....	52
Inconvenientes de la criptografía simétrica	53
4.3 EL ESTÁNDAR DE ENCRYPTADO DE DATOS (DES).....	54
Introducción a la Sección 4.3.....	54
Introducción a DES.....	54
Modos de operación de DES (I)	55
La potencia de DES.....	56
Triple DES (TDES).....	57
Otros algoritmos simétricos	58
4.4 FUNCIONES "HASH"	59
Introducción a la Sección 4.4.....	59
Definición.....	59
Funciones hash de un solo sentido (MDC).....	61
Funciones hash de un solo sentido (MAC).....	62
Requisitos de una función hash	63
Propiedades.....	64
4.5 CRIPTOGRAFÍA ASIMÉTICA	66
Introducción a la Sección 4.5.....	66
Definición de criptografía de clave pública	66
Aclaración de malentendidos en criptografía asimétrica.....	68

<i>Confidencialidad en criptografía asimétrica</i>	69
<i>Autenticación en criptografía asimétrica</i>	70
<i>Firma digital</i>	71
4.6 EL ALGORITMO DIFFIE-HELLMAN	73
<i>Introducción a la Sección 4.6</i>	73
<i>Descripción</i>	73
<i>Ataque al algoritmo Diffie-Hellman</i>	75
4.7 ALGORITMO RSA Y CERTIFICADOS DIGITALES	77
<i>Introducción a la Sección 4.7</i>	77
<i>Definición</i>	77
<i>Algoritmo RSA paso a paso</i>	78
<i>Certificados de Clave Pública</i>	80
<i>Distribución de claves y certificados</i>	82
<i>Autoridades de certificación</i>	83
5- TÉCNICAS DE AUTENTIFICACIÓN	85
5.1 TÉCNICAS DE AUTENTIFICACIÓN	86
<i>Introducción a la Sección 5.1</i>	86
<i>Definición</i>	86
<i>Tipos</i>	87
<i>Autenticación con passwords tradicionales</i>	88
5.2 PASSWORDS DE UN SOLO USO	89
<i>Introducción a la Sección 5.2</i>	89
<i>Definición. Descripción de S/Key</i>	89
<i>Un ejemplo de S/Key</i>	91
<i>Un ejemplo de S/Key (II)</i>	93
5.3 OTROS SISTEMAS DE AUTENTIFICACIÓN	94
<i>Introducción a la Sección 5.3</i>	94
<i>Introducción</i>	94
<i>Password Authentication Protocol (PAP)</i>	95
<i>Challenge Handshake Authentication Protocol (CHAP)</i>	96
<i>Ventajas de CHAP</i>	97
<i>Desventajas de PAP y CHAP</i>	98
<i>Terminal Access Controller Access-Control System (TACACS)</i>	99
<i>Ventajas e inconvenientes de TACACS</i>	100
<i>Remote Authentication Dial-In User Service (RADIUS)</i>	102
6- PROTOCOLOS DE SEGURIDAD	104
6.1 PROTOCOLOS DE SEGURIDAD.....	105
<i>Introducción a la Sección 6.1</i>	105
<i>Introducción</i>	105
6.2 PROTOCOLO SSL	107
<i>Introducción a la Sección 6.2</i>	107
<i>Definición y características</i>	107
<i>Handshake SSL</i>	109
<i>Autenticación del servidor</i>	111
<i>HTTPS (HTTP sobre SSL)</i>	113
6.3 EL PROTOCOLO DE SEGURIDAD PARA COMPRAS EN INTERNET SET	114
<i>Introducción a la Sección 6.3</i>	114
<i>Definición y características</i>	114
<i>Autenticación SET</i>	116
<i>Privacidad SET</i>	117
<i>Integridad en SET</i>	118
<i>Funcionamiento de SET</i>	119
6.4 PRETTY GOOD PRIVACY (PGP)	120
<i>Introducción a la Sección 6.4</i>	120

<i>Encriptación, compresión y firma digital usando PGP</i>	120
6.5 IPSEC.....	122
<i>Introducción a la Sección 6.5</i>	122
<i>Definición y tipos</i>	122
<i>Asociaciones de seguridad</i>	124
<i>Protocolo Authentication Header - AH (I)</i>	126
<i>Protocolo Authentication Header - AH (II)</i>	127
<i>Autenticación utilizando claves MD5</i>	128
<i>Encapsulado de seguridad de carga útil (ESP)</i>	130
<i>Envío y recepción de un paquete ESP</i>	131
<i>Modo transporte ESP</i>	133
<i>Modo túnel ESP</i>	134
<i>Comparación entre métodos</i>	135
<i>Aplicaciones ESP</i>	136
<i>Utilización de AH frente a ESP</i>	138
<i>Utilización conjunta de AH y ESP</i>	139
7- DISPOSITIVOS DE SEGURIDAD	141
7.1 FIREWALLS.....	142
<i>Introducción a la Sección 7.1</i>	142
<i>¿Qué es un firewall?</i>	142
<i>Funcionalidades de los firewalls</i>	143
<i>Características de los firewalls</i>	144
7.2 TIPOS DE FIREWALLS.....	145
<i>Introducción a la Sección 7.2</i>	145
<i>Routers que restringen paquetes o filtros de paquetes</i>	145
<i>Características del filtro de paquetes</i>	147
<i>Host bastión</i>	148
<i>Características del host bastión</i>	149
<i>DMZ o Red de Zona Perimetral</i>	151
<i>Configuración de una DMZ estándar</i>	152
<i>Servidores proxy</i>	154
<i>Ejemplo de un servidor proxy</i>	156



Bienvenido al capítulo:

Introducción a la Seguridad

1.1 Antecedentes

Introducción a la Sección 1.1

Vas a comenzar el apartado 1.1:

Antecedentes



Hasta hace unas décadas los requisitos de seguridad de la información se solventaban con procedimientos físicos y administrativos.

En las dos últimas décadas hemos asistido a dos revoluciones en estos comportamientos:

- Con la introducción de los ordenadores, fue evidente la necesidad de herramientas automáticas para proteger los ficheros y otras informaciones almacenadas en su memoria.
- Con la introducción de los sistemas distribuidos y la utilización de redes y facilidades de comunicación, se hacen necesarias medidas de seguridad en red para proteger los datos durante su transmisión y garantizar que los datos transmitidos son auténticos.

1.2 Definición y objetivos de la Seguridad

Introducción a la Sección 1.2

Vas a comenzar el apartado 1.2:

Definición



¿Y qué entendemos por seguridad?

Realmente, no existe una definición estricta de lo que se entiende por seguridad de los datos, puesto que ésta abarca múltiples y muy diversas áreas relacionadas con los sistemas de información.

Tampoco es único el objetivo de la seguridad.

Son muy diversos los tipos de amenazas contra los que debemos protegernos.

Desde amenazas físicas, como los cortes eléctricos, hasta errores no intencionados de los usuarios, pasando por los virus informáticos o el robo, destrucción o modificación de la información.

Aspectos generales



No obstante, sí hay tres aspectos fundamentales que definen la seguridad informática:

La **confidencialidad**, la **integridad** y la **disponibilidad**.

Dependiendo del tipo de sistema con el que tratemos (militar, comercial, bancario, ...), el orden de importancia de estos tres factores es diferente, e incluso entran en juego otros elementos como la autenticidad o el no repudio.

El enfoque de la política de seguridad y de los mecanismos utilizados para su implementación está influido por el más importante de los tres aspectos.

Estos aspectos también pueden entenderse como metas u objetivos.

Se entiende por **confidencialidad** el servicio de seguridad que asegura que la información no pueda estar disponible para personas, entidades o procesos no autorizados. Este aspecto de la seguridad es particularmente importante para los organismos públicos, sobre todo los de defensa.

Algunos de los mecanismos utilizados para salvaguardar la confidencialidad de los datos son, por ejemplo: el uso de técnicas de control de acceso a los sistemas y el cifrado de la información confidencial o de las comunicaciones.

Se entiende por **integridad** el servicio de seguridad que garantiza que la información es modificada, incluyendo su creación y borrado, sólo por el personal autorizado.

En el ámbito de las redes y las comunicaciones, un aspecto de la integridad es la autenticidad: proporcionar los medios para verificar que el origen de los datos es el correcto, quién los envió y cuándo fueron enviados y recibidos.

En el campo de la criptografía hay diversos métodos para mantener/asegurar la autenticidad y la integridad de los mensajes y la precisión de los datos recibidos. Uno de ellos es el uso de firmas digitales.



Se entiende por **disponibilidad** el grado en que un dato está en el lugar, momento y forma en que es requerido por el usuario autorizado. La situación que se produce cuando se puede acceder a un sistema en un periodo de tiempo considerado aceptable.

Disponibilidad significa que el sistema se mantiene funcionando eficientemente y que es capaz de recuperarse rápidamente en caso de fallo.

Lo opuesto a disponibilidad, y uno de los posibles métodos de ataque a un sistema informático, se denomina "denegación de servicio" (denial of service). Una denegación de servicio significa que los usuarios no pueden obtener del sistema los recursos deseados.

Objetivo de la seguridad



Hemos visto una definición global de seguridad. Pero, ¿cuál es el objetivo que perseguimos al implantar los servicios que mencionamos anteriormente?

Otra forma de entender la seguridad es tener en cuenta que todo sistema o información es vulnerable, es decir, susceptible a amenazas.

Una máxima de la seguridad es que: "No existe ningún sistema completamente seguro".

Existen sistemas más o menos seguros, y más o menos vulnerables, pero la seguridad nunca es absoluta.

La seguridad trata del desarrollo de contramedidas que ayuden a prever o resolver las amenazas que afectan nuestras redes, bancos de información y comunicaciones.

Vulnerabilidad

Punto o aspecto del sistema que es susceptible de ser atacado o de dañar la seguridad del mismo.

Representan las debilidades o aspectos falibles o atacables en el sistema.

Amenazas

Representan los posibles atacantes o factores que aprovechan las debilidades del sistema.

Contramedidas

Técnicas de protección del sistema contra las amenazas.



Resumen

Tal y como hemos analizado hasta el momento, la seguridad abarca cuestiones más allá de la simple ocultación de la información que transmitimos por redes consideradas inseguras.

Se ocupa también del almacenamiento seguro de la información, de la disponibilidad de los sistemas a los usuarios finales, de los planes de contingencia desarrollados para cuando fallen los sistemas, porque tenderán a fallar, del control de acceso a recursos, es decir, de analizar todas las vulnerabilidades que podamos y encontrar las medidas que hagan nuestras redes y sistemas menos vulnerables a ataques o amenazas.

Pero recuerda: *ningún sistema es completamente seguro, aún cuando implementemos toda clase de medidas de seguridad.*

Esto es, **la seguridad no es una práctica estática.**



Bienvenido al capítulo:

Aspectos de la Seguridad

2.1 Vulnerabilidades y amenazas

Introducción a la Sección 2.1

Vas a comenzar el apartado 2.1:

Clases de vulnerabilidades



En la imagen puedes ver una clasificación genérica de los diferentes tipos de vulnerabilidades a las que nuestros sistemas pueden verse expuestos.

Como observarás, la vulnerabilidad es un concepto bastante amplio, y solucionar o prever ataques a todos estos posibles puntos débiles de nuestros sistemas es una labor de titanes.

Vulnerabilidad física

Se relaciona con la posibilidad de entrar o acceder físicamente al sistema para robar, modificar o destruir el mismo.

Vulnerabilidad natural

Se refiere al grado en que el sistema puede verse afectado por desastres naturales o ambientales que pueden dañar el sistema.

Vulnerabilidad del hardware y del software

Desde el punto de vista del hardware, ciertos tipos de dispositivos pueden ser más vulnerables que otros.



Ciertos fallos o debilidades del software del sistema hacen más fácil acceder al mismo y lo hacen menos fiable.

Vulnerabilidad de los medios o dispositivos

Se refiere a la posibilidad de robar o dañar los discos, cintas, listados de impresora, etc.

Vulnerabilidad por emanación

Todos los dispositivos eléctricos y electrónicos emiten radiaciones electromagnéticas.

Vulnerabilidad de las comunicaciones

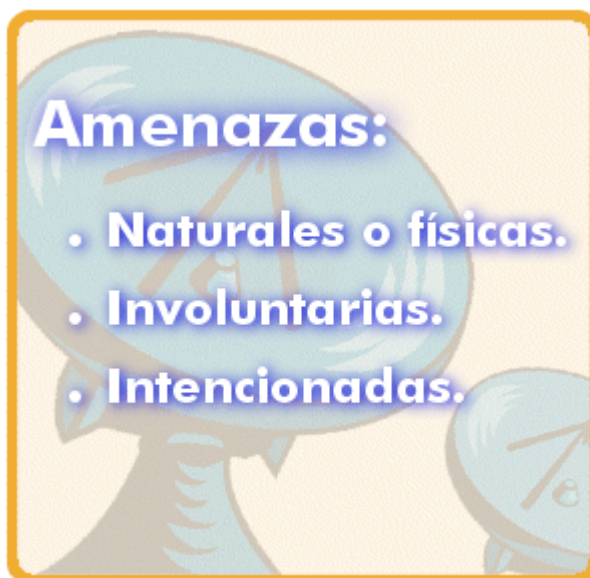
Se puede penetrar al sistema a través de la red, e interceptar información que es transmitida desde o hacia el sistema.

Vulnerabilidad humana

La gente que administra y utiliza el sistema representa la mayor vulnerabilidad del sistema.

Los usuarios del sistema también suponen un gran riesgo al mismo, el 50% de los problemas de seguridad detectados son debidos a los usuarios de los mismos.

Tipos de amenazas por su origen



Las amenazas pueden clasificarse de muy diversas formas:

Dependiendo de su origen, amenazas en redes, por su naturaleza.

Desde el punto de vista del origen o la fuente de la amenaza, podemos clasificarlas en:

- Naturales,
- involuntarias,
- intencionadas.

Amenazas naturales o físicas

Son las que ponen en peligro los componentes físicos del sistema.

Amenazas involuntarias

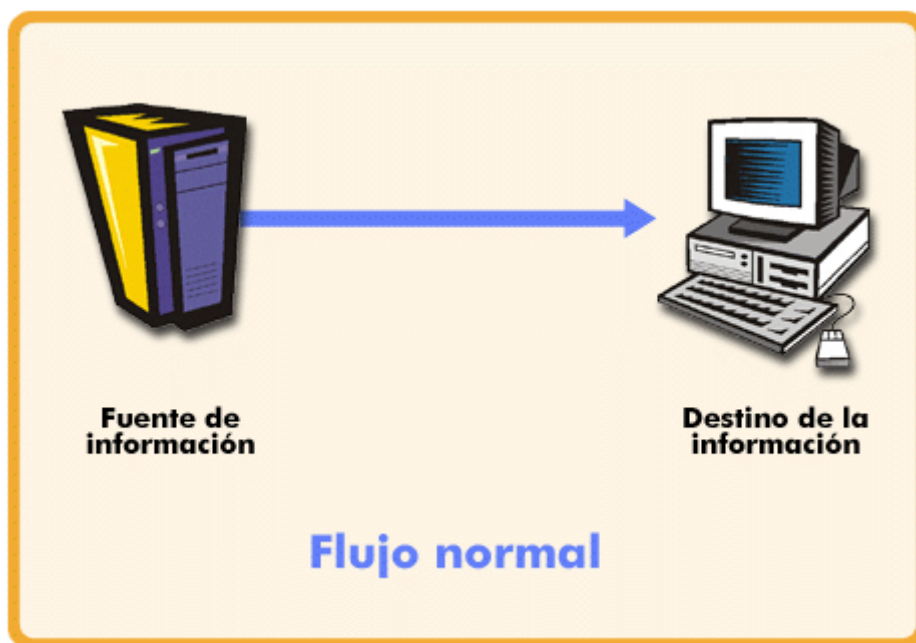
Son aquellas relacionadas con el uso descuidado del equipo, por falta de entrenamiento o de concienciación sobre la seguridad.

Amenazas intencionadas

Son aquellas procedentes de personas que pretenden acceder al sistema para borrar, modificar o robar la información, para bloquearlo o por simple diversión.

Tipos de amenazas en redes

Si hablamos de redes, los tipos de agresión se caracterizan mejor si vemos al sistema como proveedor de información.



En general, existirá un flujo de información desde un origen, como puede ser un fichero o una región de memoria principal, a un destino, como, por ejemplo, otro fichero o un usuario.

Este flujo normal se muestra en el gráfico.

Así, podemos distinguir cuatro categorías generales de agresión:

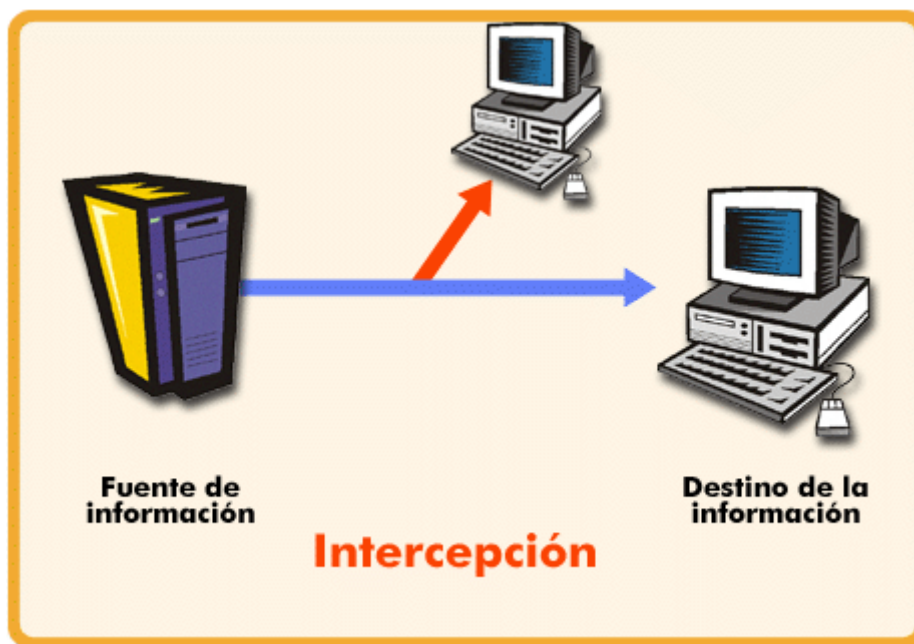
1- Intercepción

En esta amenaza, un ente no autorizado consigue acceder a un recurso.

Ésta es una agresión a la confidencialidad, ya que podemos “observar” la información intercambiada entre dos sistemas.

El ente no autorizado puede ser una persona, un programa o un ordenador.

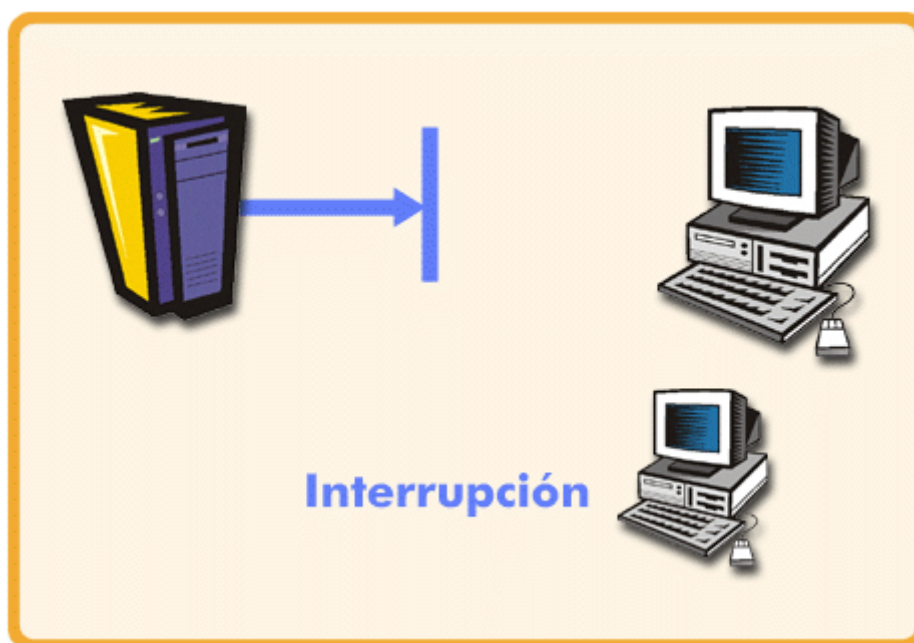
Ejemplos de agresiones a la confidencialidad son las intervenciones de las líneas para capturar datos y la copia ilícita de ficheros o datos.



2- Interrupción.

Si por causas externas un recurso del sistema se destruye o no llega a estar disponible o se inutiliza, se presenta una interrupción.

Ésta es una agresión a la *disponibilidad*.



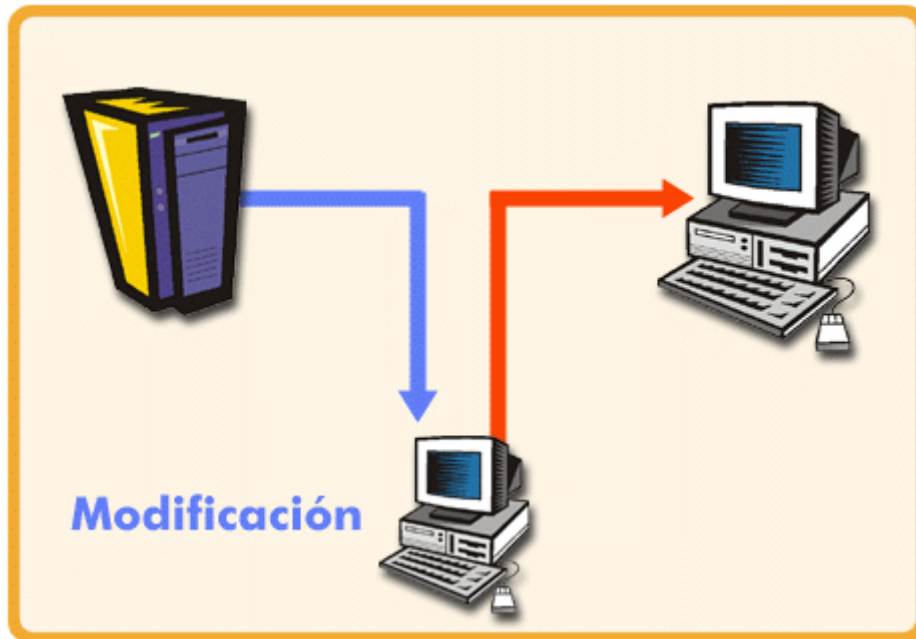
Ejemplos de este tipo de agresión son: la destrucción de un elemento hardware, como un disco duro, la ruptura de una línea de comunicación o la deshabilitación del sistema de gestión de ficheros.

En general, como veremos más adelante, se produce un ataque denominado *denial of service* (denegación de servicio).

3- Modificación

Recordemos que la Intercepción permite acceder a la información a un recurso no autorizado.

Pero si además de ganar el acceso, se deteriora el recurso o se modifica, estamos agrediendo la *integridad* del mismo.



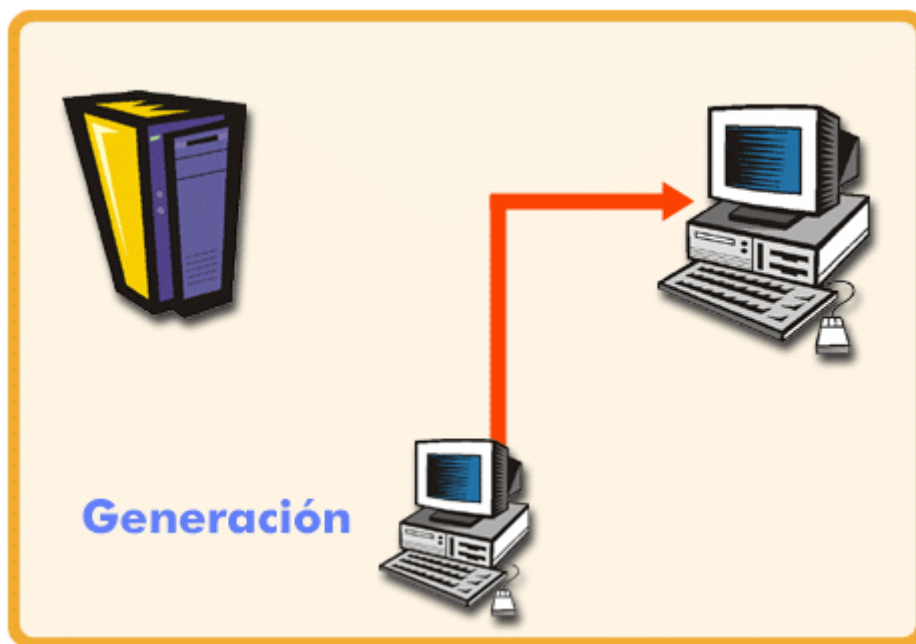
Algunos ejemplos son: los cambios de valores en un fichero de datos, alterando un programa para que funcione de una forma diferente, y modificando el contenido de los mensajes que se transmiten en una red.

4- Generación

El último tipo de agresión se presenta cuando el ente no autorizado inserta objetos falsos en el sistema.

Ésta es una agresión a la *autenticidad*.

Algunos ejemplos son la inclusión de mensajes espurios en una red o la incorporación de registros a un fichero.





Tipos de amenazas por su naturaleza

Hemos visto que las amenazas pueden ser, por su origen, naturales, voluntarias o involuntarias. También hemos descrito los tipos de amenazas a los que se ven expuestas las redes.

Ahora vamos a ver que esas mismas amenazas pueden clasificarse en dos grandes grupos en función de su naturaleza: amenazas activas o amenazas pasivas.

Amenazas pasivas



Las agresiones pasivas son del tipo de las escuchas o monitorizaciones ocultas de las transmisiones.

La meta del oponente es obtener información que está siendo transmitida.

Existen dos tipos de agresiones pasivas:

Divulgación del contenido de un mensaje o análisis del tráfico.

Las agresiones pasivas son muy difíciles de detectar, ya que no implican la alteración de los datos.

Sin embargo, es factible impedir el éxito de estas agresiones. Así, el énfasis para tratar estas agresiones está en la prevención antes que en la detección.

Divulgación del contenido de un mensaje

Una conversación telefónica, un mensaje de correo electrónico, un fichero transferido, pueden contener información sensible o confidencial.

Así, sería deseable prevenir que el oponente se entere del contenido de estas transmisiones.

Análisis de tráfico

Supón que tenemos un medio de enmascarar el contenido de los mensajes u otro tipo de tráfico de información, aunque se capturan los mensajes, no se podría extraer la información del mensaje. La técnica más común para enmascarar el contenido es el encriptado.

Pero incluso si tenemos protección de encriptado, el oponente podría ser capaz de observar los modelos de estos mensajes. El oponente podría determinar la localización y la identidad de los ordenadores que se están comunicando y observar la frecuencia y la longitud de los mensajes intercambiados. Esta información puede ser útil para extraer la naturaleza de la comunicación que se está realizando.

Amenazas activas



La segunda categoría de agresiones es la de las agresiones activas.

Estas agresiones suponen la modificación del flujo de datos o la creación de flujos falsos, y se subdividen en cuatro categorías: enmascaramiento, repetición, modificación de mensajes y denegación de un servicio.

Un **enmascaramiento** tiene lugar cuando una entidad pretende ser otra entidad diferente. Una agresión de enmascaramiento normalmente incluye una de las otras formas de agresión activa.

Por ejemplo, se puede captar una secuencia de autenticación y reemplazarla por otra secuencia de autenticación válida, así se habilita a otra entidad autorizada con pocos privilegios a obtener privilegios extras suplantando a la entidad que los tiene.

La **repetición** supone la captura pasiva de unidades de datos y su retransmisión subsecuente, para producir un efecto no autorizado.

La **modificación** de mensajes significa, sencillamente, que alguna porción de un mensaje legítimo se altera, o que el mensaje se retrasa o se reordena para producir un efecto no autorizado.

Por ejemplo, un mensaje con un significado "Permitir a Juan leer el fichero confidencial de cuentas" se modifica para tener el significado "Permitir a Pedro leer el fichero confidencial de cuentas".

La **denegación** de un servicio impide o inhibe el uso o gestión normal de las facilidades de comunicación.

Esta agresión puede tener un objetivo específico: por ejemplo, una entidad puede suprimir todos los mensajes dirigidos a un destino particular (por ejemplo, al servicio de vigilancia de seguridad).



Otro tipo de denegación de servicio es la perturbación de una red completa, deshabilitándola o sobrecargándola con mensajes, de forma que se degrade su rendimiento.

2.2 Medidas de seguridad

Introducción a la Sección 2.2

Vas a comenzar el apartado 2.2:

Definición y tipos de contramedidas



Ciertamente podemos deducir que nuestros sistemas se ven continuamente expuestos a amenazas de muy diversos tipos, dada la natural vulnerabilidad de los mismos.

¿Y qué acciones tomamos al respecto? **Lo principal es diseñar o adaptar nuestros sistemas de acuerdo a medidas de seguridad.**

Los sistemas pueden diseñarse de acuerdo con criterios de economía, de eficiencia y de eficacia, etc., porque son claramente medibles y se asocian a parámetros que, maximizando unos y minimizando otros, se puede tender hacia diseños óptimos.

Diseñar sistemas mediante criterios de seguridad es más complejo, pues las amenazas son en muchos casos poco cuantificables y muy variadas.

La aplicación de medidas para proteger el sistema supone un análisis y cuantificación previa de los riesgos o vulnerabilidades del sistema, y la definición de una política de seguridad y su implementación a través de una serie de medidas.

Las medidas de seguridad que pueden establecerse en un sistema informático son de cuatro tipos fundamentales: lógicas, físicas, administrativas y legales.

Vamos a verlas con más detalle.

Medidas físicas



Lo primero que tenemos que hacer es aplicar mecanismos para impedir el acceso directo o físico no autorizado al sistema.

También debemos proteger al sistema de desastres naturales o condiciones medioambientales adversas.

Se trata fundamentalmente de establecer un perímetro de seguridad en nuestro sistema.

Para ello, tendremos en cuenta:

- El acceso físico al sistema por parte de personas no autorizadas.
- Los daños físicos por parte de agentes nocivos o contingencias.
- Las medidas de recuperación en caso de fallo.

Medidas lógicas

Una vez solventado el problema de las vulnerabilidades físicas y naturales, debemos preocuparnos por las medidas de acceso a los recursos y a la información y al uso correcto de los mismos, así como a la distribución de las responsabilidades entre los usuarios.



Nos referimos concretamente a la protección de la información almacenada o transmitida.

¿Quieres conocer algunas medidas de tipo lógico?

De eso trata la mayor parte de este curso.

Algunas medidas lógicas

Entre los tipos de controles lógicos que es posible incluir en una política de seguridad podemos destacar los siguientes:

- ☒ Establecimiento de una política de **control de accesos**. Incluyendo un sistema de identificación y autenticación de usuarios autorizados y un sistema de control de acceso a la información, por ejemplo el uso de identificadores y passwords (palabras de paso).
- ☒ Definición de una **política de intalación y copia de softwre** (sistemas de backup).
- ☒ Uso de la **criptografía** para proteger los datos y las comunicaciones.
- ☒ Uso de **cortafuegos** para proteger una red local de Internet.



Definición de una **política de copias de seguridad**.



Definición de una **política de monitorización (logging) y auditoría (auditing)** del sistema.

Dentro de las medidas lógicas se incluyen también aquellas relativas a las personas y que podríamos denominar medidas humanas. Se trata de definir las funciones, relaciones y responsabilidades de distintos usuarios potenciales del sistema

Medidas administrativas



Llegados a este punto, pensaríamos que hemos hecho todo lo posible por solventar las vulnerabilidades de nuestro sistema.

Sin embargo, debemos recordar el factor humano. Para ello, existen las medidas administrativas y legales.

Las medidas administrativas son aquellas que deben ser tomada por las personas encargadas de definir la política de seguridad para ponerla en práctica, hacerla viable y vigilar su correcto funcionamiento.

- Documentación y publicación de la política de seguridad y de las medidas tomadas para ponerla en práctica.
- Debe quedar claro quién fija la política de seguridad y quién la pone en práctica.
- Establecimiento de un plan de formación del personal.

Los usuarios deben tener los conocimientos técnicos necesarios para usar la parte del sistema que les corresponda. Este tipo de conocimiento es fundamental para evitar toda una serie de fallos involuntarios que pueden provocar graves problemas de seguridad.

Medidas legales



Las medidas legales se utilizan para disuadir al posible atacante o para aplicarle algún tipo de castigo a posteriori.

A diferencia de las administrativas, éste tipo de medidas trascienden el ámbito de la empresa y normalmente son fijadas por instituciones gubernamentales e incluso instituciones internacionales.

Un ejemplo de este tipo de medidas es la **LORTAD** (Ley Orgánica de Regulación del Tratamiento Automatizado de Datos de Carácter Personal).

Esta ley vincula a todas las entidades que trabajan con datos de carácter personal, define las medidas de seguridad para su protección y las penas a imponer en caso de su incumplimiento.

Resumen del Reglamento de Seguridad de la LORTAD

Reglamento de Seguridad de la LORTAD.

Establece la obligación de implantar diferentes medidas de seguridad en los sistemas informáticos que alberguen datos personales, así como realizar una auditoría informática bianual, en determinados casos.

Ámbito de aplicación.

Establecer las medidas de índole técnica y organizativas necesarias para garantizar la seguridad que deben reunir los ficheros automatizados, los centros de tratamiento, locales, equipos, sistemas, programas y las personas que intervengan en el tratamiento automatizado de los datos de carácter personal sujetos al régimen de la LORTAD.

Niveles de seguridad: bajo, medio y alto.

Nivel medio: Afecta a los ficheros que contengan datos relativos a la comisión de infracciones administrativas o penales, Hacienda Pública, servicios financieros y servicios de información sobre solvencia patrimonial y crédito.

Medidas de seguridad de nivel medio.

1. Documento de seguridad.
2. Régimen de funciones y obligaciones del personal.
3. Registro de incidencias
4. Identificación y autenticación de usuarios.
5. Control de acceso.
6. Gestión de soportes.
7. Copias de respaldo y recuperación.
8. Responsable de seguridad.
9. Auditoría bianual.
10. Medidas adicionales de identificación y autenticación.
11. Control de acceso físico.
12. Medidas adicionales de gestión de soportes.
13. Registro de incidencias.
14. Pruebas sin datos reales.

Plan de Contingencia

Aunque ya hemos definido los tipos de contramedidas de seguridad (físicas, lógicas, administrativas y legales), no podemos olvidarnos de que al hablar de políticas de seguridad hay que contemplar tanto la prevención como la recuperación.



La mayor parte de las medidas de las que hemos hablado hasta este momento se refieren a la prevención ante posibles amenazas.

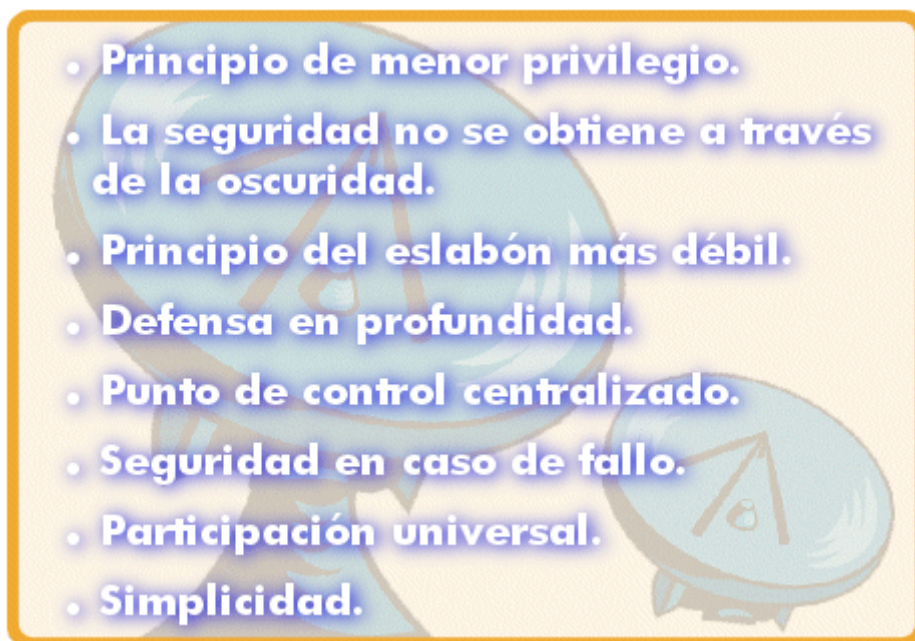
Sin embargo, y como ya hemos comentado anteriormente, ningún sistema es completamente seguro, y por tanto hay que definir una estrategia a seguir en caso de fallo o desastre.

De hecho los expertos de seguridad afirman sutilmente que **hay que definir un plan de contingencia para cuando falle el sistema, no por si falla el sistema.**

Existen otros aspectos relacionados con la recuperación como son la detección del fallo, la identificación del origen del ataque y de los daños causados al sistema y la toma de medidas a posteriori contra el atacante.

Todo ello se basa en buena medida en el uso de una adecuada política de monitorización y auditoría del sistema.

Principios fundamentales de la Seguridad



Para finalizar este capítulo, recordemos los principios que rigen la seguridad informática.

Principio de menor privilegio

Cualquier objeto debe tener sólo los privilegios de uso necesarios para desarrollar su tarea, y ninguno más.

La seguridad no se obtiene a través de la oscuridad

Un sistema no es más seguro porque escondamos sus posibles defectos o vulnerabilidades, sino porque los conozcamos y corriamos, estableciendo las medidas de seguridad adecuadas.

Principio del eslabón más débil

En un sistema de seguridad, el atacante siempre acaba encontrando y aprovechando los puntos débiles o vulnerabilidades.

Defensa en profundidad

La seguridad de nuestro sistema no debe depender de un mecanismo, sino que es necesario establecer varios mecanismos sucesivos.

Punto de control centralizado

Se trata de establecer un único punto de acceso a nuestro sistema, de modo que cualquier atacante que intente acceder al mismo tenga que pasar por él.

Seguridad en caso de fallo

Este principio afirma que en caso de que cualquier mecanismo de seguridad falle, nuestro sistema debe quedar en un estado seguro.



Participación universal

La participación voluntaria de todos los usuarios en la seguridad de un sistema es el mecanismo más fuerte conocido para hacerlo seguro.

Simplicidad

Mantener las cosas lo más simples posibles.

La complejidad permite esconder múltiples fallos.



Resumen

Recordemos algunos términos que utilizaremos de aquí en adelante.

Las vulnerabilidades de un sistema lo hacen susceptible a cuatro tipos fundamentales de amenazas:

- Interrupción.
- Intercepción.
- Modificación.
- Generación.

De éstas, sólo la intercepción es de tipo pasivo, siendo difícilmente detectable. Las otras tres conllevan por defecto la modificación o destrucción de la información o el servicio.

Para que nuestros sistemas sean menos vulnerables, implementamos medidas de seguridad de tipo físico, lógico, administrativo y legal.



Bienvenido al capítulo:

Seguridad en Redes IP

3.1 Seguridad en redes IP

Introducción a la Sección 3.1

Vas a comenzar el apartado 3.1:

Definición



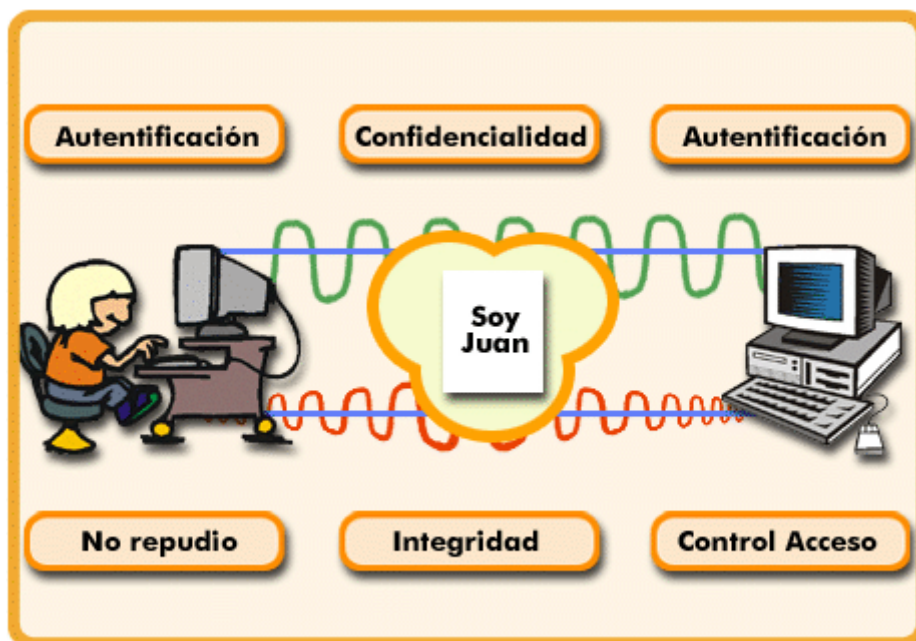
A partir de ahora, empezaremos a introducir definiciones y conceptos mucho más orientados a las redes IP.

Para comenzar, veamos cómo se define la seguridad en redes IP:

Dentro del modelo de referencia OSI, se define una arquitectura de seguridad ("*Information Processing Systems. OSI Reference Model - Part 2: Security Architecture*", ISO/IEC IS 7498-2).

De acuerdo con esta arquitectura, para proteger las comunicaciones de los usuarios a través de una red, es necesario dotar a las mismas con una serie de servicios, que se conocen como servicios de seguridad.

Servicios de seguridad en redes IP



Autenticación de la entidad par

Este servicio verifica la fuente de los datos.

La autenticación puede ser sólo de la entidad origen, de la entidad destino o de ambas a la vez.

Confidencialidad de los datos

Este servicio evita que se revelen, deliberada o accidentalmente, los datos de una comunicación.

Autenticación de la entidad par

Este servicio verifica la fuente de los datos.

La autenticación puede ser sólo de la entidad origen, de la entidad destino o de ambas a la vez.

No repudio (irrenunciabilidad)

Este servicio proporciona la prueba, ante una tercera parte, de que cada una de las entidades ha participado, efectivamente, en la comunicación.

Puede ser de dos tipos:

- Con prueba de origen o emisor: el destinatario tiene garantía de quién es el emisor concreto de los datos.
- Con prueba de entrega o receptor: el emisor tiene prueba de que los datos de la comunicación han llegado íntegramente al destinatario correcto en un instante dado.



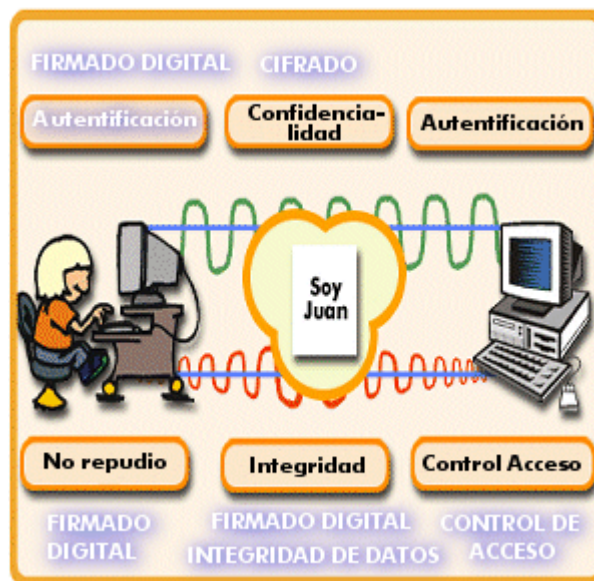
Integridad de los datos

Este servicio verifica que los datos de una comunicación no se alteren, esto es, que los datos recibidos por el receptor coincidan por los enviados por el emisor.

Control de acceso

Este servicio verifica que los recursos son utilizados por quien tiene derecho a hacerlo.

Mecanismos de seguridad



Hasta este momento, básicamente es la misma definición de seguridad que contemplamos en los capítulos anteriores. Ten en cuenta que también hablamos de autenticación y no repudio, además de confidencialidad, integridad y control de acceso.

¿Qué debemos hacer para proporcionar los citados servicios?

Para ello es necesario incorporar en los niveles adecuados del modelo de referencia OSI los mecanismos de seguridad que ves en la imagen.

➡ Cifrado.

El cifrado puede hacerse mediante el uso de criptosistemas simétricos o asimétricos y puede aplicarse extremo a extremo o a cada enlace del sistema de comunicaciones.

➡ Firmado digital.

La firma digital se puede definir como un conjunto de datos que se añaden a una unidad de datos de modo que protejan a ésta contra cualquier falsificación, permitiendo al receptor comprobar el origen y la integridad de los datos. Para ello, se cifra la unidad de datos junto con alguna componente secreta del firmante, y se obtiene un valor de control ligado al resultado cifrado.

Para que se pueda proporcionar el servicio de no repudio con prueba de entrega, hay que forzar al receptor para que envíe un acuse de recibo firmado digitalmente.

➡ Control de acceso.

Se usa para autenticar las capacidades de una entidad para acceder a un recurso dado. El control de acceso se puede llevar a cabo en el origen o en un punto intermedio, y se encarga de asegurar que el emisor está autorizado a comunicarse con el receptor o a usar los recursos de comunicación.



➡ Integridad de datos

Hay que distinguir entre la integridad de una unidad de datos individual y la integridad de una secuencia de unidades de datos.

Para lograr integridad de una unidad de datos, el emisor añade datos suplementarios a la unidad de datos. Estos datos suplementarios se obtienen en función de la unidad de datos y, generalmente, se cifran. El receptor genera los mismos datos suplementarios a partir de la unidad original y los compara con los recibidos.

Para proporcionar integridad a una secuencia de unidades de datos se requiere, adicionalmente, algún mecanismo de ordenación, tal como el uso de números de secuencia, un sello temporal o un encadenamiento criptográfico entre las unidades.

➡ Intercambio de autenticación.

Tiene dos grados:

- Autenticación simple: el emisor envía su identificador y una contraseña al receptor, el cual los comprueba.
- Autenticación fuerte: utiliza propiedades de los criptosistemas de clave pública.

El **mecanismo de firmado digital** soporta los servicios de integridad de los datos, autenticación del emisor y no repudio con prueba de origen.

El **mecanismo de cifrado** soporta el servicio de confidencialidad de los datos.

El **mecanismo de intercambio de autenticación** soporta el servicio de autenticación de entidad par.

El **mecanismo de integridad de datos** soporta el servicio de integridad de datos.

El **control de acceso** soporta el servicio de control de acceso.

3.2 Amenazas a la Seguridad en redes IP

Introducción a la Sección 3.2

Vas a comenzar el apartado 3.2:

Introducción



Cuando nos referimos a los entornos de las redes IP, la seguridad, tanto de los propios datos como de las comunicaciones, depende de algunos de los factores anteriormente mencionados: integridad, autenticación y confidencialidad.

Desafortunadamente, el diseño original de los protocolos TCP/IP y las redes que utilizan estos protocolos, como Internet, no permite asegurar que estas tres características que definen la seguridad en la transmisión de datos puedan garantizarse. En ausencia de las medidas de seguridad adecuadas, las transmisiones en las redes IP pueden estar sujetas a una amplia gama de amenazas.

Vamos a revisar los tipos más comunes de amenazas antes de pasar a explicar en detalle los diferentes tipos de soluciones que se presentan para combatirlas.

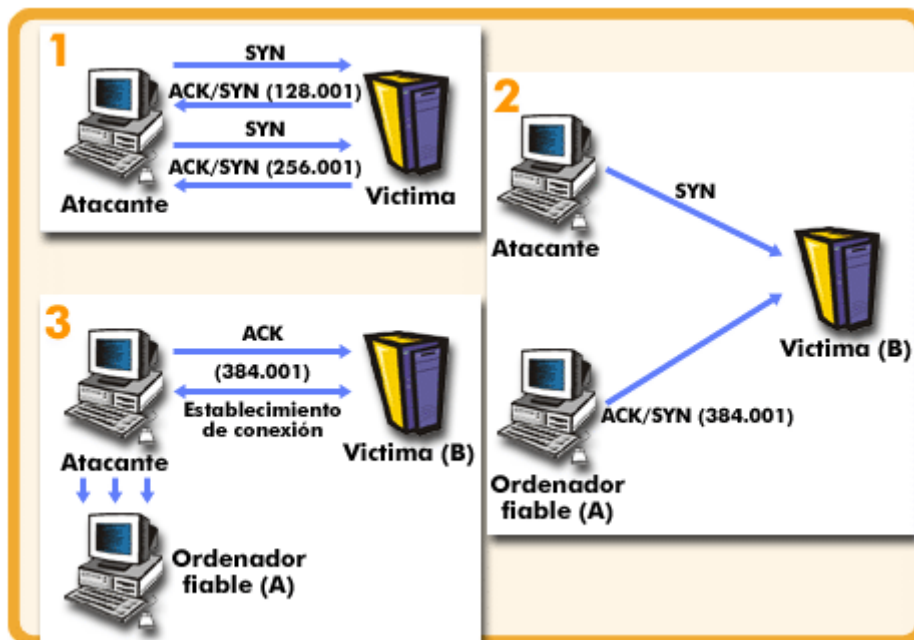
Autenticación significa que la persona con quien nos estamos comunicando es, realmente, quien dice ser; es un paso más allá de la identificación, porque se verifica dicha identificación.

Mantener la **confidencialidad** en las comunicaciones equivale a asegurar que nadie puede espiar los datos que se transfieren en dichas comunicaciones; es decir, que nadie puede leer los datos incluso aunque los intercepte.

Garantizar la **integridad** de los datos significa que los datos no pueden ser alterados de ninguna forma, sin que el destinatario se dé cuenta, durante la transmisión.

Spoofing o suplantación

Las técnicas de Spoofing intentan sacar partido del hecho de que un atacante puede utilizar la dirección IP de alguien y pretender presentarse frente a un interlocutor con la dirección IP que no le corresponde.



Esto es para obtener información del comportamiento de la víctima (generación de números de secuencia, etc.).

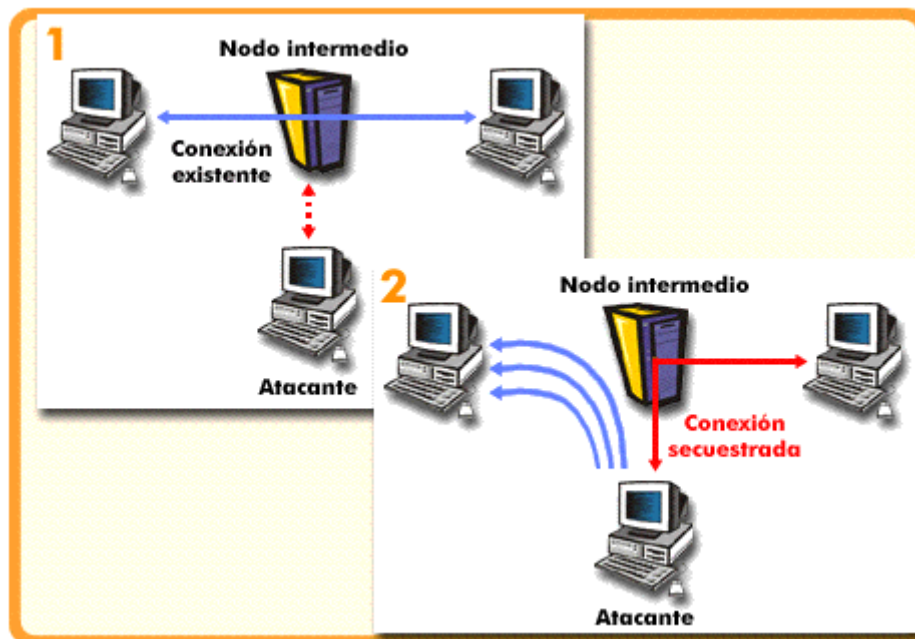
La técnica de Spoofing suele crear el malentendido común de que puede utilizarse para navegar en Internet, conversar en línea, enviar correo y cosas por el estilo.

Eso, generalmente, no es cierto.

La suplantación o falsificación de la dirección IP hace que las respuestas sean dirigidas hacia el ordenador suplantado, en vez de hacia el atacante, lo que quiere decir que el atacante no crea una conexión de red "normal".

Sin embargo, dicha alteración es parte integral de muchos ataques a redes que no necesitan ver las respuestas (intercepción ciega).

Hijacking o secuestro de sesión



Las técnicas de spoofing no suelen presentarse en solitario, sino que constituyen un primer nivel de ataque que posibilita otros tipos de ataques posteriores.

En el "Secuestro de la Sesión" o Session Hijacking, lo que se intenta es "robar" una conexión ya existente entre dos ordenadores, en vez de suplantar la identidad de uno de ellos para crear una nueva conexión.

Electronic Eavesdropping (sniffing)



Existe un tipo de software, conocido coloquialmente por sniffer.

Este tipo de herramientas capturan todo el tráfico de red que pasa a su través, constituyéndose en una herramienta fundamental para los administradores de redes Ethernet para la detección de problemas, permitiéndoles determinar, con rapidez, lo que está sucediendo en cualquier segmento de red.

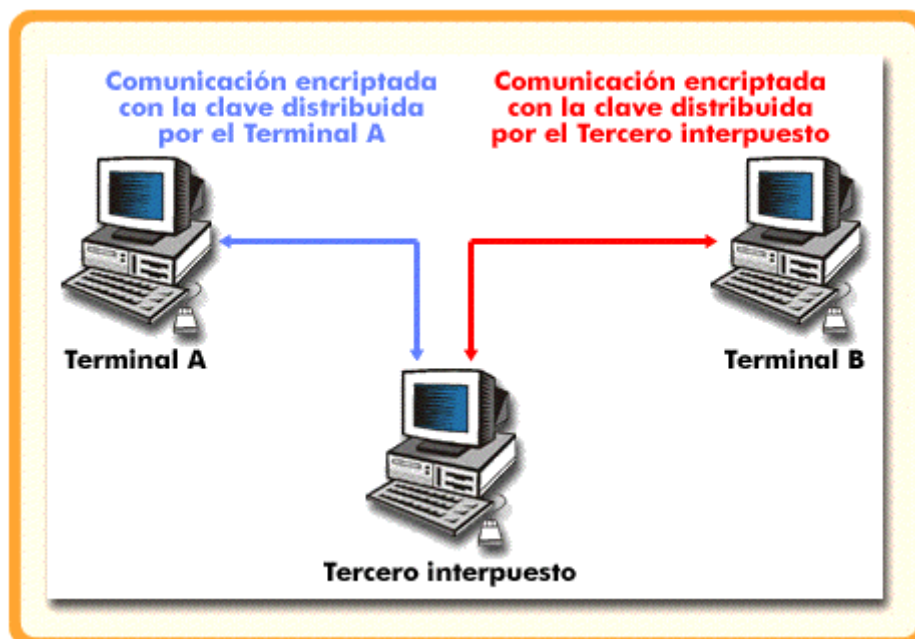
Sin embargo, en manos de alguien que pretenda escuchar comunicaciones privadas, un sniffer es una potente herramienta de espionaje.

Por ejemplo, un atacante podría utilizar un sniffer de paquetes para capturar todos los paquetes de login a una red y, después, utilizar la información de login para entrar a sistemas a los que, de otro modo, no tendría un acceso autorizado.

Las técnicas de sniffing también pueden utilizarse para capturar datos de corporaciones y mensajes que se transmitan por la red, para su análisis posterior.

Por ejemplo, el atacante puede realizar un análisis de tráfico para aprender quién se comunica con quién.

The man in the middle



Un atacante experto, empleando la suplantación (spoofing), secuestro (hijacking) y espionaje (sniffing), podría utilizar estas técnicas durante la fase de intercambio de claves y hacerse con la clave de encriptación de un sistema.

(No te preocupes, luego veremos la encriptación en detalle).

Acto seguido, podría sembrar por toda la red su propia clave de encriptación, de forma que mientras un usuario piensa que se está comunicando utilizando la clave proporcionada por el otro extremo, en realidad estará usando la clave conocida por el "tercero interpuesto".

En resumen, la particularidad de este ataque radica en que, en el modelo cliente-servidor, el cliente piensa que se está comunicando con el servidor y viceversa, cuando, en realidad, los dos se comunican con el tercero interpuesto sin saberlo y, además, utilizando la clave de encriptación propuesta por dicho tercero.



Resumen

Como has podido observar, la transmisión de datos sobre redes IP se ve expuesta a amenazas que mezclan varios de los tipos genéricos descritos en el capítulo anterior: generación, modificación, interrupción e interceptación.

Por ello, se definen una serie de servicios de seguridad, basados en los parámetros generales de confidencialidad, integridad y disponibilidad.

Estos servicios son ofrecidos mediante mecanismos de seguridad como la criptografía, el firmado digital, el control de acceso y las técnicas de autenticación.

A partir de este momento, vamos a adentrarnos en cada uno de estos mecanismos y ver las funcionalidades que nos ofrecen.



Bienvenido al capítulo:

Criptografía



4.1 Introducción

Introducción a la Sección 4.1

Vas a comenzar el apartado 4.1:

Definición

Los modernos algoritmos criptográficos de hoy día junto con los potentes microprocesadores disponibles en el mercado hacen posible la utilización masiva de métodos potentes de autenticación y encriptación.

¿Qué entendemos por criptografía?

Bajo el término “criptografía” se sitúan un gran número de algoritmos para la encriptación y desencriptación de la información, clasificados de acuerdo a:

- La forma en que los secretos, o claves, son compartidos entre los comunicantes.
- Cómo se utilizan los secretos para encriptar y desencriptar la información.
- Cómo son los algoritmos.

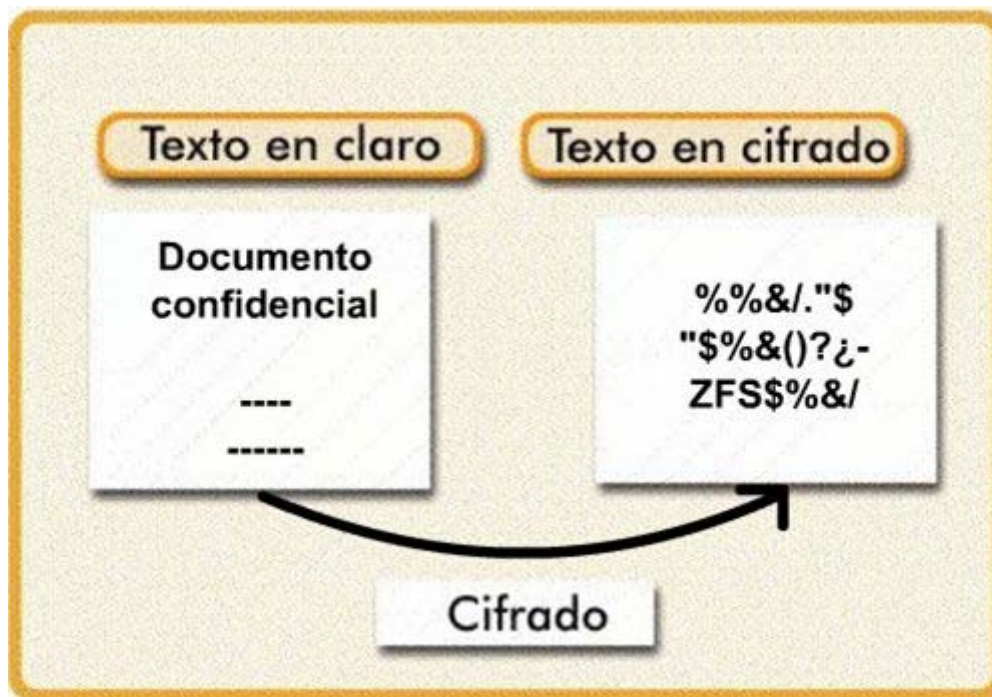
A continuación, te vamos a mostrar unos pocos algoritmos criptográficos que son particularmente útiles para la seguridad de las redes y las VPNs.

¿Qué es encriptación?

Bueno, la encriptación o codificación de la información para evitar que ésta sea leída por partes no autorizadas ha sido la principal utilización de la criptografía desde tiempos muy remotos.

Por ejemplo, ya Julio César en la época romana utilizaba un método de este tipo para transmitir mensajes a sus comandantes durante las batallas.

Algoritmos y claves de cifrado



En primer lugar, te recuerdo en qué se basa la encriptación:

Para que la encriptación funcione correctamente, ambas partes, emisor y receptor, deben conocer el mismo conjunto de reglas, llamadas "cifrador", que se han utilizado para transformar la información original en su forma codificada, llamada a menudo "texto cifrado".

Por ejemplo, un "cifrador" simple puede añadir un número arbitrario de caracteres, digamos 13, a todos los caracteres del mensaje.

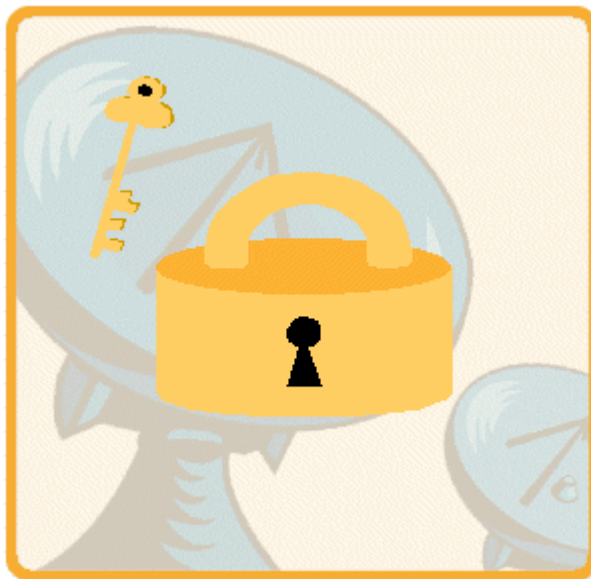
En cuanto la parte receptora conozca lo que hizo el emisor con el mensaje, podrá invertir el proceso (por ejemplo, eliminando 13 caracteres del mensaje recibido) para extraer el texto original.

Pensarás: ya, ya, pero ¿y lo del algoritmo y la clave?

Pues son los dos componentes de la encriptación: un algoritmo criptográfico es una función matemática que combina texto no cifrado ("texto en claro"), u otra información inteligible, con una cadena de dígitos, denominada clave, para producir un texto cifrado ininteligible.

Ambos elementos, clave y algoritmo, son cruciales en el proceso de encriptación.

Ventajas de los algoritmos de cifrado con claves



¿Y cuáles son las ventajas que nos ofrecen los métodos de encriptación que hacen uso de claves?

Hombre, pues fundamentalmente dos.

Los algoritmos de encriptación son difíciles de diseñar; no podríamos crear un nuevo algoritmo cada vez que quisiéramos comunicarnos en privado con un nuevo participante. Si utilizamos claves podremos utilizar el mismo algoritmo para comunicarnos con mucha gente; hay que utilizar una clave diferente con cada participante.

Si alguien intercepta los mensajes encriptados y es capaz de descubrir la clave de encriptación, basta con cambiar dicha clave de nuevo para volver a encriptar los mensajes con la nueva clave; no necesitamos cambiar de algoritmo (a menos que se haya demostrado que es el algoritmo, y no la clave, el que es inseguro; esto no suele suceder).



La longitud de las claves

Pero, por desgracia, el número de claves disponibles no será infinito.



Bueno, pues el número disponible depende del número de bits en la clave. Por ejemplo, una clave de 8 bits de longitud sólo permite 256 (2^8) combinaciones, o claves, posibles. Cuanto mayor sea el número de claves posibles, más difícil será “romper” un mensaje encriptado.

Por tanto, el nivel de dificultad depende de la longitud de la clave. A un ordenador normal, no le llevaría mucho tiempo intentar secuencialmente cada una de las 256 claves posibles (menos de un milisegundo) y desencriptar un mensaje para ver si tiene sentido o no. Pero si se utiliza una clave de 100 bits, lo que equivale a tener que buscar entre 2^{100} claves diferentes, y el ordenador puede comprobar un millón de claves por segundo, emplearía varios siglos en descubrir la clave verdadera.

Resumiendo, para que lo entiendas bien, la seguridad de un algoritmo de encriptación está relacionada con la longitud de su clave, porque conociendo que una clave tiene una longitud de n bits, se tiene una idea de cuánto tiempo se debe emplear para romper el código.



Véras ahora claramente que la seguridad del algoritmo depende exclusivamente del secreto de la clave, pero se te escapará por qué no hacer también inaccesible el algoritmo.



Bueno, si la seguridad dependiese de aspectos tales como el secreto del algoritmo, o de la inaccesibilidad al texto cifrado, sin utilizar por tanto clave alguna, personas no autorizadas podrían obtener esa información a partir de publicaciones especializadas o podrían llegar a capturar tráfico en teoría inaccesible para ellos, teniendo así fácil el acceso a la información protegida.

¿Y esa es la única razón?, te preguntarás.

Bueno, también el hecho de que no necesitemos mantener en secreto el algoritmo, significa que los fabricantes pueden y tienen desarrolladas implementaciones de los algoritmos de encriptado en circuitos integrados de bajo coste. Estos circuitos están ampliamente difundidos, y están incorporados en un gran número de productos, haciéndolos muy accesibles.

Espero que ahora te haya quedado más claro, pero me imagino que también verás que existirá algún problema.

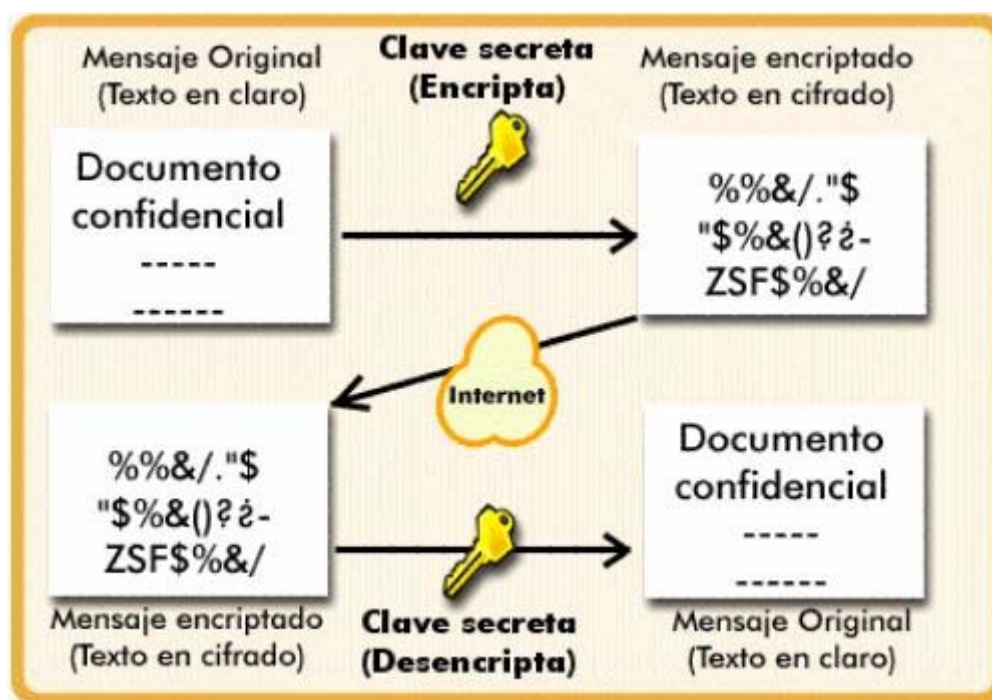
Sí, el principal problema de seguridad con el uso del encriptado convencional es mantener en secreto la clave.

4.2 Criptografía simétrica

Introducción a la Sección 4.2

Vas a comenzar el apartado 4.2:

Definición



Habrás oído que existen dos tipos de criptografías: de clave privada y de clave pública; ¿qué te puedo contar de ello?

Bueno, vayamos por partes: la forma más antigua de criptografía basada en claves se **denomina encriptación de clave privada o simétrica**. En este esquema, ambos emisor y receptor poseen la misma clave, lo que significa que ambas partes pueden encriptar y desencriptar los mensajes con la clave secreta.

¿Y qué ventajas proporciona este tipo de criptografía?

Las ventajas de la utilización de la criptografía de clave simétrica es en primer lugar la existencia de algoritmos muy rápidos y eficientes, especialmente si se implementan en hardware.

Además, si la longitud de la clave es lo bastante larga (típicamente se usan valores de 56 a 128 bits), es casi imposible reventarlas usando la fuerza bruta.



Inconvenientes de la criptografía simétrica

¿Y además no tiene ningún tipo de desventaja?



Bueno, pues por desgracia no: por ejemplo, ambas partes deben estar de acuerdo en el secreto compartido, es decir, en la clave. Si tenemos varias comunicaciones con otras tantas partes, tenemos que mantener a la vez varias claves secretas, una por cada una de las partes contrarias, ya que si utilizaremos la misma clave con más de una parte contraria, una de ellas podría leer los mensajes que estamos enviando a otra de las partes.

También, este tipo de criptografía presenta problemas con la autenticidad, porque la identidad del generador o del receptor del mensaje no puede probarse, ya que en este caso tanto transmisor como receptor de los mensajes utilizan la misma clave, con lo que ambos pueden crear y encriptar mensajes y asegurar que es la otra persona la que los envía.

De esta manera, nunca se podrá tener la certeza de que alguien en concreto de los que posee la clave secreta es el que ha enviado un determinado mensaje, ya que también podría haberlo hecho cualquiera de los otros poseedores de la misma clave.

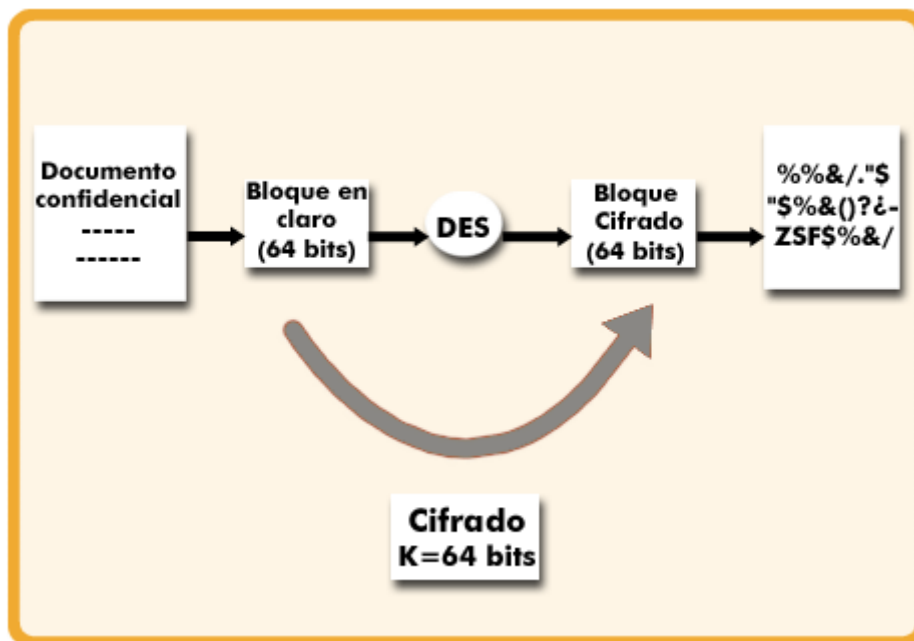
Pero como estarás interesado en ver mecanismos en concreto de criptografía, te enseño a continuación los principales algoritmos de encriptación mediante clave secreta.

4.3 El estándar de encriptado de datos (DES)

Introducción a la Sección 4.3

Vas a comenzar el apartado 4.3:

Introducción a DES



DES significa Data Encryption Standard, y es un algoritmo desarrollado originalmente por IBM a requerimiento del NBS de EE.UU (National Bureau of Standards, en la actualidad denominado NIST, National Institute of Standards and Technology) y posteriormente modificado y adoptado por el gobierno de EE.UU. en 1977 como estándar de cifrado de todas las informaciones sensibles no clasificadas.

En un principio trabajaba sobre bloques de 128 bits, teniendo la clave igual longitud, pero tras las modificaciones introducidas por el NBS, consistentes básicamente en la reducción de la longitud de claves y bloques, DES cifra bloques de 64 bits cada vez, produciendo así 64 bits cifrados.

Modos de operación de DES (I)



¿A que no te imaginabas que existieran varios modos de operación DES?, bueno, pues existen cuatro modos de operación definidos para DES (los cuales han sido generalizados para cualquier cifrador de bloque).

Estos modos fueron estandarizados en 1980 por NIST y se trata de los que ves en la imagen.

ECB (Electronic CodeBook)

Cifra cada bloque de 64 bits del mensaje en claro, uno tras otro con la misma clave de 56 bits.

Un par de bloques idénticos de mensaje en claro producen bloques idénticos de mensaje cifrado.

CBC (Cipher Block Chaining)

Sobre cada bloque de 64 bits del mensaje en claro se ejecuta un OR exclusivo con el bloque previo del mensaje cifrado, antes de proceder al cifrado con la clave DES.

De este modo, el cifrado de cada bloque depende del anterior, y bloques idénticos de mensaje en claro producen diferentes mensajes cifrados.

CFB (Cipher FeedBack)

El cifrado de un bloque de mensaje en claro procede de ejecutar un OR exclusivo del bloque de mensaje en claro con el bloque previo cifrado.

CFB puede modificarse para trabajar con bloques de longitud inferior a 64 bits.

OFB (Output FeedBack)

Similar al modo CFB, excepto en que los datos sobre los que se ejecuta el OR exclusivo junto con los bloques de mensaje en claro, es generada independientemente del mensaje en claro y del mensaje cifrado.



La potencia de DES



¿Qué exigencias se le ha hecho a DES, sobre todo desde su adopción como estándar federal americano?

Naturalmente ha habido una preocupación persistente sobre el nivel de seguridad proporcionado por DES. La preocupación mas seria hoy en día es el tamaño de la clave. Con una longitud de 56 bits, existen 2^{56} claves posibles, lo que es aproximadamente $7,6 \times 10^{16}$ claves. Con un método a base de fuerza bruta, es decir, probando todas las 2^{56} posibles claves se ha podido romper DES en Enero de 1999.



Lo anterior quiere decir que, es posible verificar todas las claves posibles en el sistema DES en un tiempo corto, lo que lo hace inseguro para propósitos de alta seguridad, es decir, ya no sirve a sus propósitos iniciales... ¡Mala suerte!

Bueno, ¿qué posibilidades nos quedan ahora? La opción que se ha tomado para poder suplantar a **DES** ha sido usar lo que se conoce como cifrado múltiple, es decir aplicar varias veces el mismo algoritmo para fortalecer la longitud de la clave. Así, ha surgido un nuevo sistema de cifrado que se conoce actualmente como **triple-DES o TDES**, y que vamos a ver a continuación...

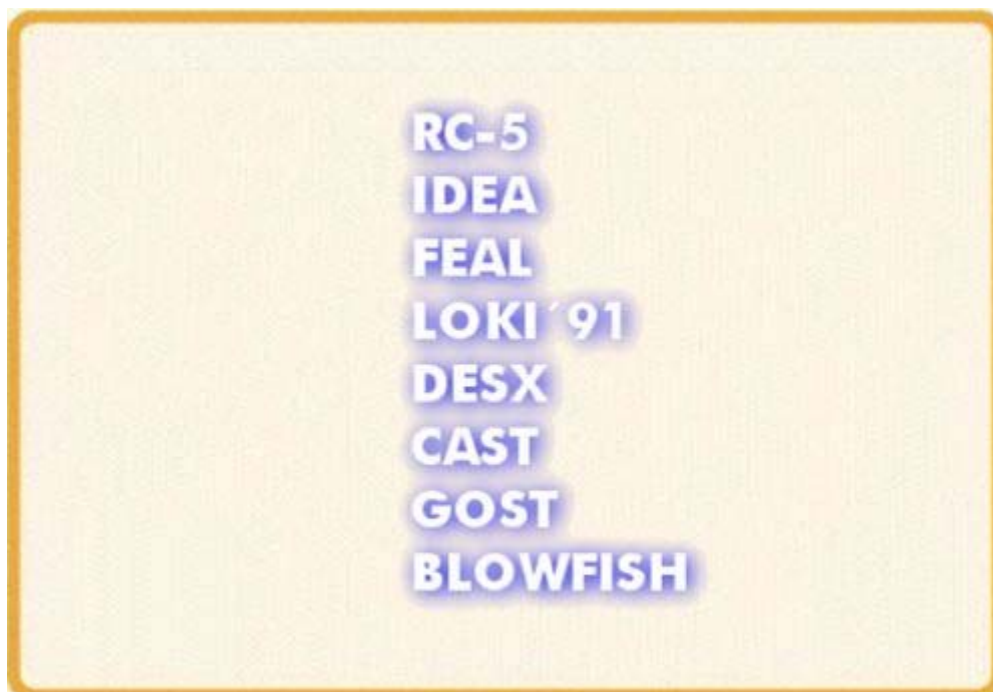
Triple DES (TDES)


$$C = EK_2[DK_1[EK_1[P]]]$$

Ya en 1979, mucho antes del ataque definitivo, IBM se dio cuenta de que la longitud de la clave DES era demasiado corta y diseñó una manera de aumentarla efectivamente utilizando codificación triple, cuya primera normalización para aplicaciones comerciales resultó ser el Triple DES.

El Triple DES utiliza dos claves y tres ejecuciones del algoritmo DES (), y la función sigue una secuencia encriptado-desencriptado-encriptado (EDE), cuya mágica fórmula es la que ves en la imagen.

Otros algoritmos simétricos



De entre todos los algoritmos de clave simétrica, se optó por TDES ya que es muy fácil interoperar con **DES** y proporciona seguridad a mediano plazo.

Pero me imagino que no te creerás que DES y TDES son los únicos algoritmos de clave simétrica existentes.

En los últimos 20 años se han diseñado una gran cantidad de sistemas criptográficos simétricos, entre algunos de ellos están: **RC-5, IDEA, FEAL, LOKI '91, DESX, Blowfish, CAST, GOST**, etcétera.

Sin embargo no han tenido el alcance de DES, a pesar de que algunos de ellos tienen mejores propiedades.

Todo esto en cuanto al pasado, pero ¿qué nos depara el futuro?

Podemos decir que el estado actual de la criptografía simétrica es la búsqueda de un nuevo sistema que pueda reemplazar a **DES** en la mayor parte de aplicaciones.

Es así como se ha optado por convocar un concurso de sistemas criptográficos simétricos, y que éste decida cuál será el nuevo estándar al menos para los próximos 20 años.

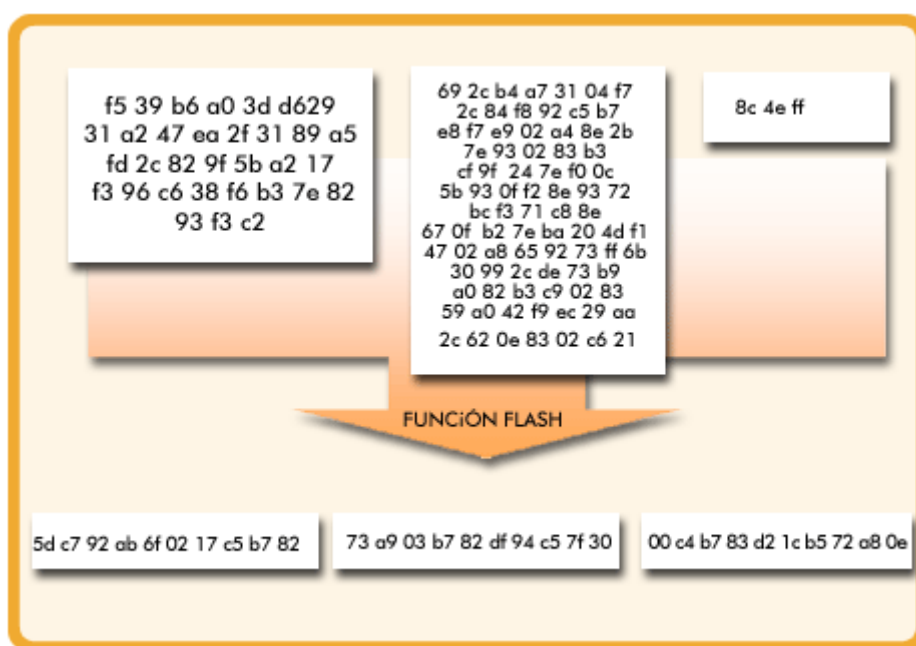
4.4 Funciones "hash"

Introducción a la Sección 4.4

Vas a comenzar el apartado 4.4:

Definición

¿Te suena el concepto de función "hash" o resumen?



Una función hash acepta un mensaje de longitud variable M como entrada y produce como salida una etiqueta única de tamaño fijo $H(M)$, llamada algunas veces resumen del mensaje.

En realidad hay un tipo principal de funciones hash.

La función hash de un solo sentido, o función hash segura, es importante no sólo para la autenticación de mensajes sino para las firmas digitales, como veremos más adelante.

¿Y cuál es el funcionamiento general de una función hash?

Todas las funciones hash operan utilizando los mismos principios generales:

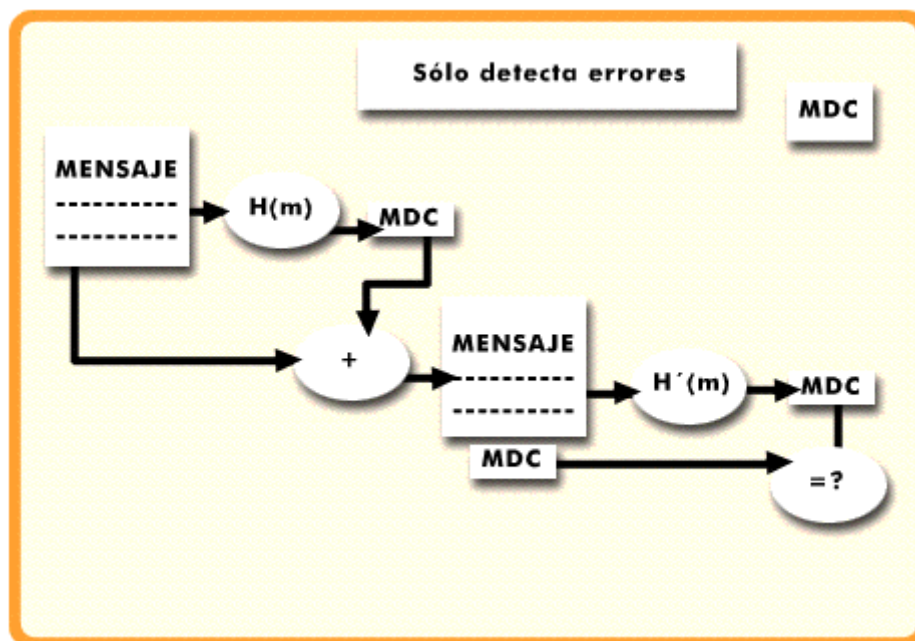
- La entrada se ve como una secuencia de bloques de n bits.
- La entrada se procesa bloque a bloque en una forma iterativa para producir un valor de la función hash de n bits.

¿Qué tal algún dato concreto?



Un ejemplo de función hash segura es el algoritmo **MD5**, que procesa en bloques de 512 bits y produce un resumen del mensaje de 128 bits.

Funciones hash de un solo sentido (MDC)



¿Cuáles son las funciones hash de un solo sentido?

Pues básicamente hay dos tipos. ¿Empezamos por el primero de ellos?

Se llama **MDC** (**M**odification **D**etection **C**odes).

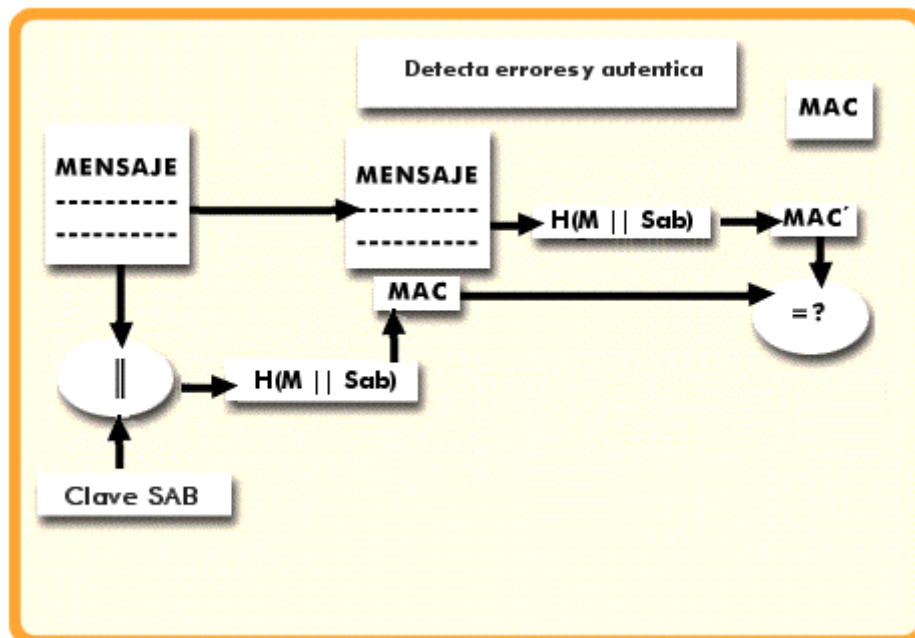
Los MDC sirven para resolver el problema de la integridad de la información; al mensaje se le aplica un MDC (una función hash) y se manda junto con el propio mensaje, al recibirlo el receptor aplica la función hash al mensaje y comprueba que sea igual al hash que se envió antes.

Es decir, se aplica un hash al mensaje **M** y se envía con el mensaje [**M**, **H(M)**]; cuando se recibe se le aplica una vez más el hash (ya que **M** es público) obteniendo **H'(M)**, si **H(M)=H'(M)**, entonces se acepta que el mensaje sea transmitido sin alteración.

Funciones hash de un solo sentido (MAC)

¿Y cuál es el segundo tipo existente?

Pues es conocido como **MAC** (**M**essage **A**uthentication **C**odes).



Los MAC sirven para autenticar los mensajes (junto con la integridad).

Esta técnica supone que las dos partes comunicantes, digamos A y B, comparten un valor secreto común SAB.

Cuando A envía un mensaje a B, calcula la función hash de la concatenación de la clave secreta con el mensaje M: $MAC = H(SAB || M)$, donde "||" significa concatenación.

Entonces, envía $[M || MAC]$ a B. Como B posee la clave SAB, puede recalcular $H(SAB || M)$ y verificar MAC, con lo que se comprueba la integridad de la clave privada SAB, demostrando que es el origen A el que manda el mensaje (autenticidad).

Como el valor secreto no se envía, no es posible que un agresor modifique un mensaje interceptado.

Mientras el valor secreto permanezca oculto, no es posible que un agresor genere un mensaje falso.

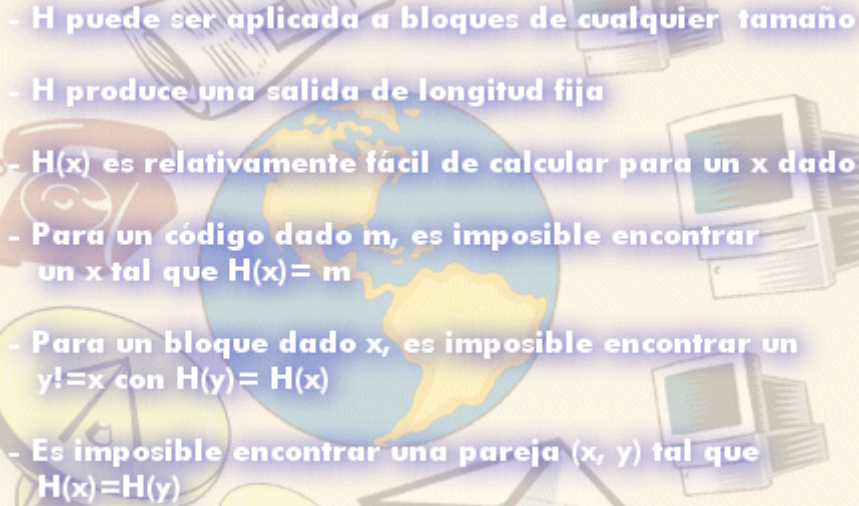
Requisitos de una función hash



El objetivo de una función hash es producir una "huella dactilar" de un fichero, un mensaje u otro bloque de datos.

Para que sea útil para autenticación, una función hash H debe tener una serie de propiedades.

- H puede ser aplicada a bloques de cualquier tamaño.
- H produce una salida de longitud fija.
- $H(x)$ es relativamente fácil de calcular para un x dado, haciendo práctica la implementación hardware y software.
- Para un código dado m , es imposible, computacionalmente, encontrar un x tal que $H(x) = m$.
- Para un bloque dado x , es imposible, computacionalmente, encontrar un $y \neq x$ con $H(y) = H(x)$.
- Es imposible, computacionalmente, encontrar una pareja (x, y) tal que $H(x) = H(y)$.



- H puede ser aplicada a bloques de cualquier tamaño

- H produce una salida de longitud fija

- $H(x)$ es relativamente fácil de calcular para un x dado

- Para un código dado m , es imposible encontrar un x tal que $H(x)=m$

Esta propiedad es importante si la técnica de autenticación supone el uso de un valor secreto (ver). El valor secreto no se envía; sin embargo, si la función hash no es de un solo sentido, un agresor puede descubrir fácilmente el valor secreto: si el agresor puede observar o interceptar una transmisión, obtiene el mensaje M y el código hash $MAC = H(SAB || M)$. El agresor entonces invierte la función hash para obtener $SAB || M$. Como el agresor tiene ahora M y $SAB || M$, es una cuestión trivial obtener SAB . Lo que se le pide a la función hash es que al aplicar la función inversa, no se obtenga el resultado original.

- Para un bloque dado x , es imposible encontrar un $y \neq x$ con $H(y) = H(x)$

Esta propiedad garantiza que no se puede encontrar un mensaje alternativo que produzca el mismo valor que un mensaje dado.

Si esta propiedad no fuera válida, un agresor sería capaz de realizar la siguiente secuencia: primero, observar o interceptar un mensaje M mas su código hash encriptado MAC; segundo, generar un código hash descriptado a partir del mensaje M ; tercero, generar un mensaje alternativo M' con el mismo código hash.

Una función hash que satisface las cinco primeras propiedades se le conoce como una función hash débil.

- Es imposible encontrar una pareja (x, y) tal que $H(x) = H(y)$

Esta propiedad protege de una clase de agresión sofisticada conocida como agresión de cumpleaños.

Si satisface también la sexta propiedad, entonces se le conoce como una función hash fuerte.

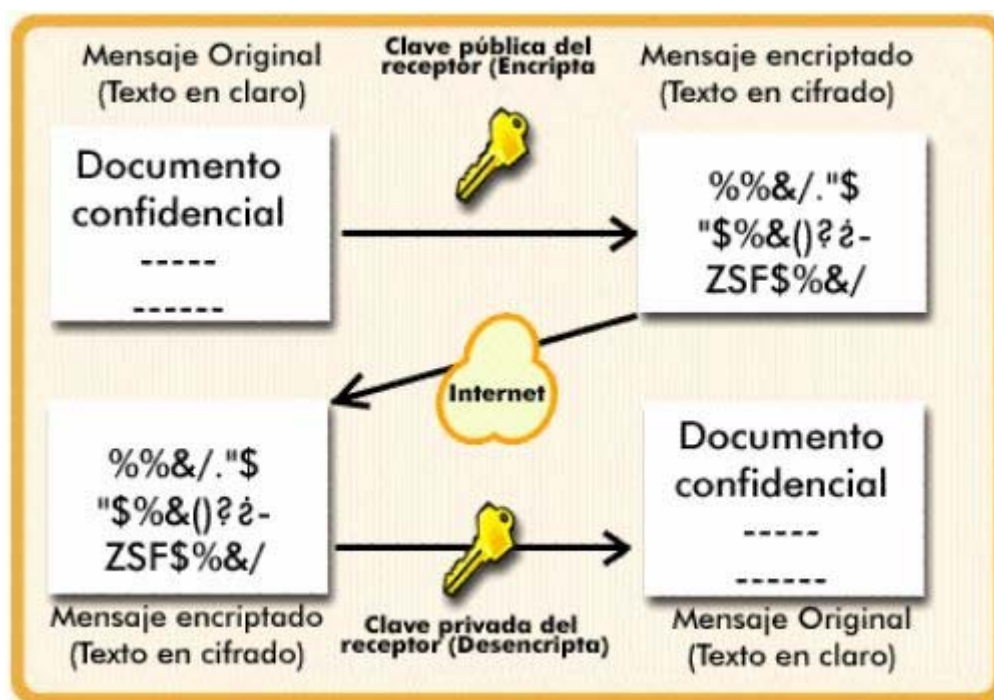
Además de proporcionar autenticación (en el caso MAC), un resumen del mensaje proporciona también integridad de los datos. Lleva a cabo la misma función que la secuencia de comprobación de trama: si se altera algún bit accidentalmente en el tránsito, el resumen del mensaje producirá error.

4.5 Criptografía asimétrica

Introducción a la Sección 4.5

Vas a comenzar el apartado 4.5:

Definición de criptografía de clave pública



Bueno, no creas que me había olvidado del otro tipo existente de criptografía, la criptografía asimétrica o de clave pública: ¿en qué consiste y qué la diferencia de la simétrica o de clave privada?

La criptografía de clave pública se basa en el concepto del par de claves. Una parte del par de claves, la "clave privada", sólo es conocida por el propietario designado; la otra parte, la "clave pública", puede difundirse a todo el mundo manteniéndose asociada a su propietario.

Los pares de claves presentan una característica única: los datos encriptados con una de las claves pueden desencriptarse con la otra y no son independientes entre sí.

¿Cuáles son las principales características de los algoritmos de clave pública?

- No es factible computacionalmente determinar la clave de desencriptado conociendo solamente el algoritmo de criptografía y la clave de encriptado.
- Cualquier clave, de las dos, que se utilizan, se puede utilizar para el encriptado y la otra para el desencriptado (aunque esto último sólo se cumple en el algoritmo RSA).



4- Criptografía - 4.5 Criptografía asimétrica

A través de las pantallas que siguen, descubriremos la potencia de este nuevo método.

¡Ah! Un último comentario: este tipo de criptografía ha sido el primer avance realmente revolucionario en el encriptado en miles de años.



Aclaración de malentendidos en criptografía asimétrica

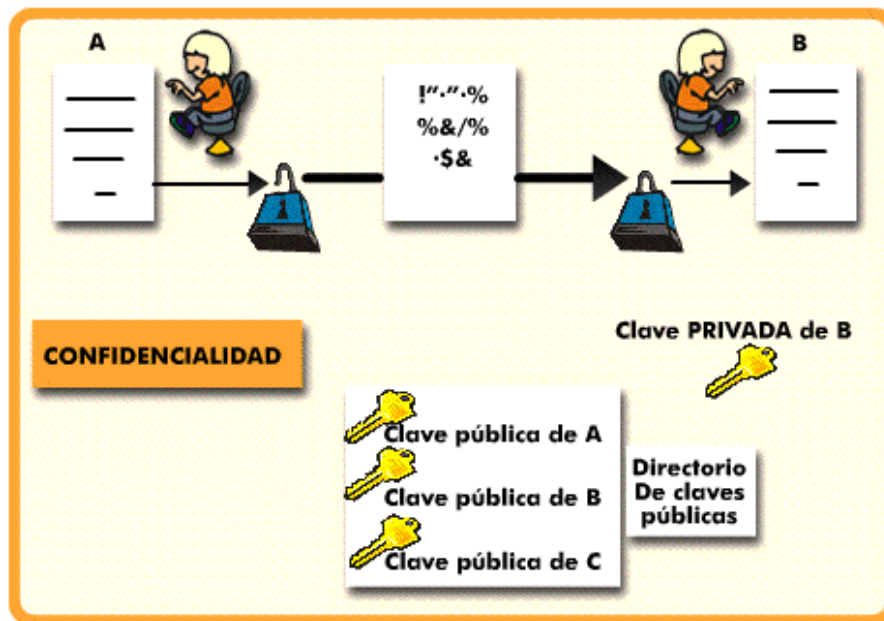
Pero no creas que este revolucionario método es la panacea de la criptografía. Algunas aclaraciones:

- ✎ No hay nada que haga suponer que el encriptado de clave pública es más seguro frente al criptoanálisis que el de clave privada. La seguridad de cualquier esquema de encriptado depende de la longitud de la clave y del trabajo computacional que requiere romper un cifrado.
- ✎ El encriptado de clave pública no ha vuelto obsoleto el encriptado de clave privada. Por el contrario, a causa de la computación suplementaria de los esquemas actuales de cifrado por clave pública, no es probable que el encriptado de clave privada sea abandonado.
- ✎ No es cierto que la distribución de claves sea trivial con el encriptado de clave pública, comparado con el diálogo que se requiere con los centros de distribución de claves en el encriptado de clave privada.

Si analizamos paso a paso los algoritmos de clave pública, vemos que el procedimiento siempre es el siguiente:

- Cada sistema final genera el par de claves que se van a utilizar para encriptar los mensajes que se van a emitir y desencriptar los que se reciban.
- Cada sistema publica su clave de encriptado. Esta es la clave pública. La clave compañera se mantiene privada.
- Si A desea enviar un mensaje a B, encripta el mensaje utilizando la clave pública de B.
- Cuando B recibe el mensaje, lo desencripta utilizando la clave privada de B. Ningún otro destino puede desencriptar el mensaje ya que solamente B y nadie más que B conoce la clave privada.

Confidencialidad en criptografía asimétrica



¿Y en cuántas formas diferentes pueden utilizarse los algoritmos de clave pública? Pues básicamente de dos formas distintas:

- Para proporcionar confidencialidad al mensaje.
- Y para probar la autenticidad del originador del mensaje.

Pero no nos aceleremos y vayamos paso a paso. En primer lugar, ¿qué es eso de la confidencialidad?

En este caso, el emisor utiliza la clave pública del receptor para encriptar un mensaje, de forma que será confidencial hasta que sea decodificado por el receptor que posee la clave privada, que es la única capaz de desencriptarlo.

Por ejemplo, si se desea mantener confidencial un mensaje, Pablo debe adquirir primero la clave pública de Inés. Después utiliza dicha clave para encriptar el mensaje y se lo envía a Inés.

Como el mensaje fue encriptado con la clave pública de Inés, sólo alguien que posea la clave privada de Inés (y presumiblemente sólo la tiene la propia Inés) podrá desencriptar el mensaje.

Por otro lado, aunque la encriptación de un mensaje con la parte pública del par de claves no difiere mucho de la encriptación con clave secreta, los sistemas de clave pública presentan algunas ventajas.

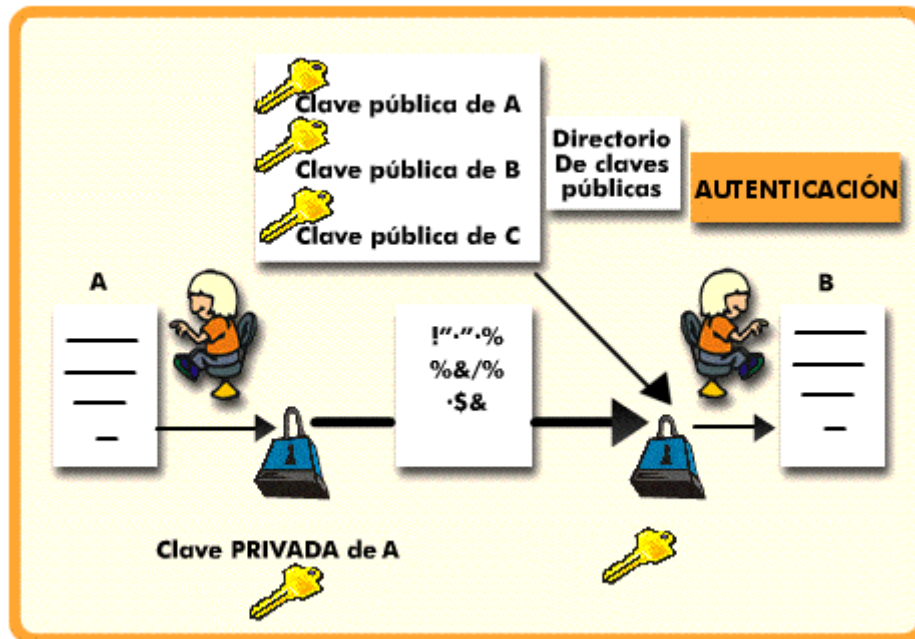
Por ejemplo, la parte pública del par de claves puede distribuirse fácilmente (por ejemplo, desde un servidor) sin miedo a que esto comprometa el uso de la clave privada.

No tenemos que enviar una copia de nuestra clave pública a todas nuestras partes contrarias; ellas mismas pueden obtenerla desde un servidor mantenido por nuestra corporación, o por un proveedor de servicios.

Autenticación en criptografía asimétrica

¿Te imaginas que puede ser la autenticación?

La idea básica es: *como nosotros somos los únicos que sabemos cómo encriptar algo con nuestra clave privada, cualquiera que utilice nuestra clave pública para desencriptar un mensaje puede estar seguro de que el mensaje viene de nosotros.*



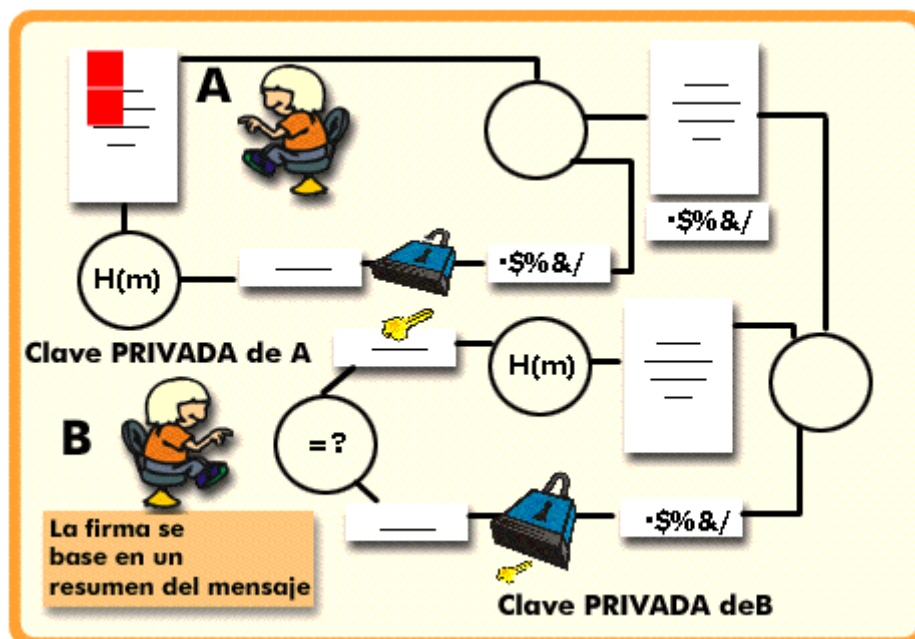
Por lo tanto, para autenticarse, un emisor encripta un mensaje con su clave privada, a la que sólo él tiene acceso.

Este mensaje sólo podrá ser desencriptado con la clave pública del emisor, probando así su origen de manera fehaciente.

Así que ten cuidado, porque la utilización de nuestra clave privada en un documento electrónico es similar a nuestra firma en el documento en papel.

De esta forma, el receptor se asegurará de que el mensaje viene de nosotros, **pero no puede asegurar que nadie más lo va a leer**, ya que la clave pública se encuentra a disposición de todo el mundo...

Firma digital



Y de la cualidad de la autenticación que proporciona la criptografía asimétrica, a la definición de la firma digital, sólo hay un paso, ¿no crees?

El problema de la utilización de algoritmos criptográficos de clave pública para autenticar mensajes es que computacionalmente son lentos; pero afortunadamente a los criptógrafos se les ha ocurrido una forma rápida de generar una representación única y abreviada de nuestro mensaje llamada "**resumen del mensaje**" que, por tanto, puede ser encriptada con rapidez y utilizada como **firma digital**.

Estos algoritmos rápidos ya los conoces, ya que son las funciones hash. Este tipo de función no utiliza una clave; se trata de una simple fórmula que convierte un mensaje de cualquier longitud en una cadena simple de dígitos llamada resumen del mensaje.

Cuando se utiliza una función hash de 16 bytes, el texto procesado con dicha función producirá una salida de 16 bytes de longitud.

Por ejemplo, un mensaje completo puede reducirse a "CBBV235ndsAG3D67".

Lo importante con las funciones *hash* es que cada mensaje produce un resumen del mensaje aleatorio.

Los resúmenes de mensajes pueden servir por sí mismos como indicador de que el mensaje no ha sido alterado, pero las firmas digitales son aún más fiables, y es que si encriptamos el resumen del mensaje con nuestra clave privada, tenemos una firma digital.

Quizás veas más claro el proceso entero de la firma digital en la figura. En ella, el emisor, Pablo, calcula un resumen del mensaje para ese mensaje, encripta dicho resumen con su clave privada y envía la firma digital, junto con el mensaje sin cifrar, hacia Inés.



A continuación, Inés utiliza la clave pública de Pablo para descryptar la firma digital, que Pablo calculó. Como Inés consigue descryptar el resumen del mensaje con la clave pública de Pablo, reconoce que fue Pablo el que creó este mensaje, autenticando al emisor. Inés utiliza la misma función hash, que fue acordada con anterioridad, para calcular el resumen del mensaje de texto en claro de Pablo.

Si el valor calculado por Inés coincide con el que Pablo envió encriptado, Inés puede asegurar que la firma digital es auténtica, lo que significa que fue Pablo el que envió el mensaje y que éste no ha sido alterado por nadie.

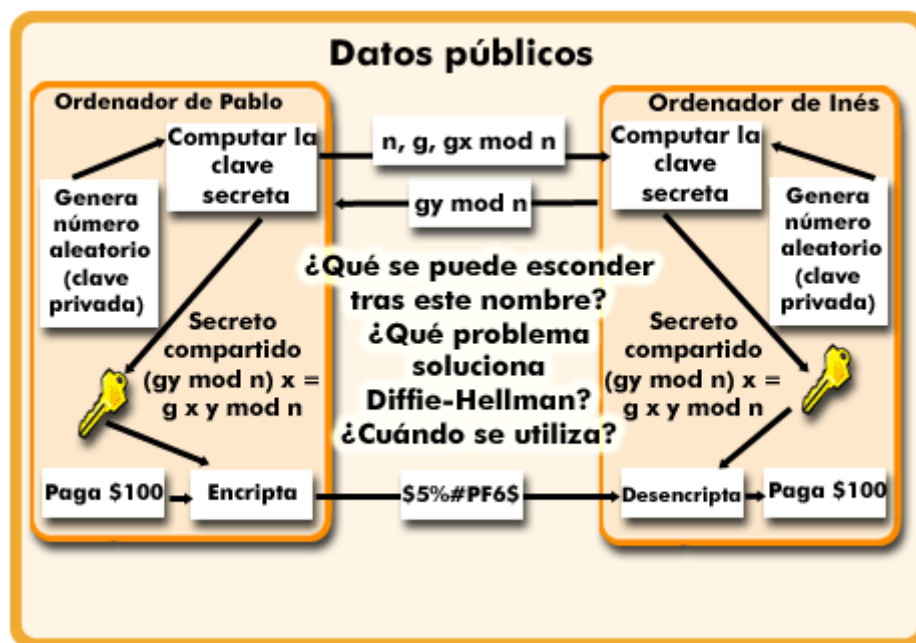
¡Todo perfecto!, ¿no? Pues tenemos la mala suerte de que no... El único problema de la autenticación con firmas digitales es que la copia de texto en claro se envía como parte del mensaje y, por tanto, no se mantiene la privacidad (es decir, alguien podría leer los datos, aunque no pueda alterarlos). Si queremos mantener la privacidad de los datos, deberíamos encriptar el mensaje, pero utilizando un algoritmo simétrico con clave privada, para reducir la sobrecarga computacional. Este procedimiento complica aún más las cosas, pero el esfuerzo puede merecer la pena, como veremos a continuación...

4.6 El algoritmo Diffie-Hellman

Introducción a la Sección 4.6

Vas a comenzar el apartado 4.6:

Descripción



¿Qué se puede esconder tras este nombre?

Pues un elemento bastante importante: se suele considerar al algoritmo Diffie-Hellman como el sistema de clave pública más antiguo (Diffie y Hellman, 1976), y ha sido muy utilizado para la gestión de claves.

Y ya veremos que las propuestas para el intercambio de claves para IPsec están basadas en el algoritmo de Diffie-Hellman.

¿Qué problema soluciona Diffie-Hellman?

Diffie-Hellman se presenta como una solución al problema de cómo dos entidades pueden ponerse de acuerdo en la utilización de un secreto, utilizando para ello canales públicos.

Sin embargo, Diffie-Hellman no soporta ni la encriptación ni las firmas digitales. Entonces, sin estas características tan importantes, ¿para qué sirve?

Creo que lo veremos más claramente con un ejemplo:

Pablo e Inés tienen que acordar dos números primos grandes, n y g , donde $(n-1)/2$ también es primo, y g debe cumplir ciertas condiciones. Estos números pueden ser públicos, por lo que cualquiera de ellos puede escoger n y g y decírselo al otro.

abiertamente. Ahora, Pablo escoge un número grande (digamos de 512 bits), x , y lo mantiene en secreto. Igualmente, Inés escoge un número secreto grande y .



Pablo inicia el protocolo de intercambio de claves enviando a Inés un mensaje que contiene $(n, g, gx \bmod n)$, como se muestra en la imagen. Inés responde enviando a Pablo un mensaje que contiene $g^y \bmod n$. Ahora Pablo toma el número que Inés le envió y lo eleva a la potencia x para obtener $(g^y \bmod n)^x$. Inés lleva a cabo una tarea parecida para obtener $(g^x \bmod n)^y$. Por las leyes de la aritmética modular, ambos cálculos arrojan $g^{xy} \bmod n$. Ahora, Pablo e Inés comparten una clave secreta $g^{xy} \bmod n$.

Cualquiera podría interceptar los valores públicos, así que sería posible para un atacante obtener g y n . Si el atacante pudiera calcular x e y , podría averiguar la clave secreta. El problema es que, dado sólo $g^x \bmod n$, no es posible encontrar x . No se conoce un algoritmo práctico para calcular logaritmos discretos módulo un número primo muy grande.

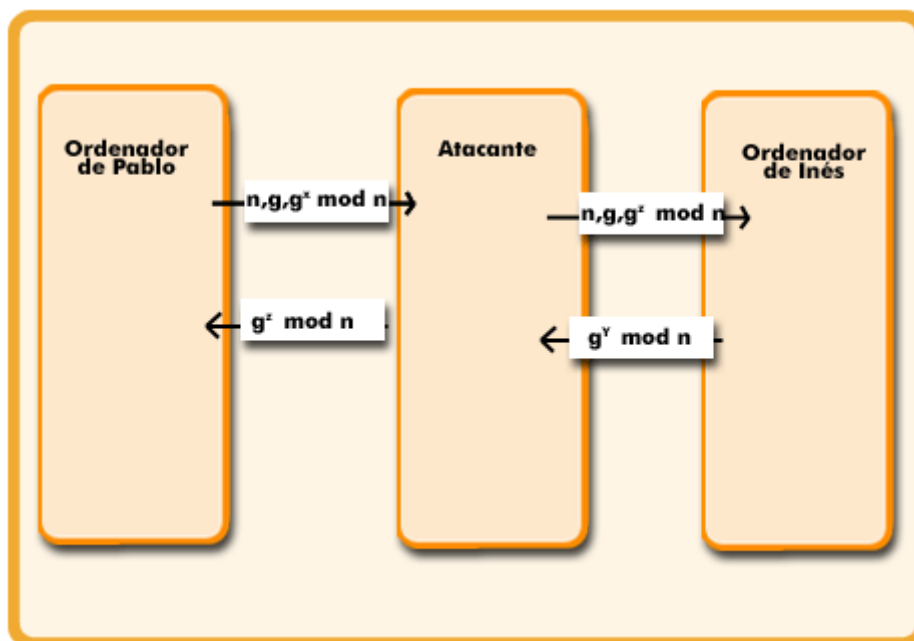
Para hacer algo más concreto el ejemplo anterior, usaremos los valores (completamente irreales) de $n = 47$ y $g = 3$. Pablo selecciona $x = 8$ e Inés selecciona $y = 10$. Ambos se mantienen en secreto. El mensaje de Pablo a Inés es $(47, 3, 28)$ porque $3^8 \bmod 47$ es 28. El mensaje de Inés a Pablo es (17) . Pablo calcula $17^8 \bmod 47$, que es 4. Inés calcula $28^{10} \bmod 47$, que es 4. Pablo e Inés han determinado independientemente que la clave secreta ahora es 4. Un atacante tendría que resolver la ecuación $3^x \bmod 47 = 28$, lo que puede lograrse mediante una búsqueda exhaustiva en el caso de números pequeños como este, pero no cuando todos los números tienen cientos de bits de longitud. Todos los algoritmos actualmente conocidos, simplemente, tardan demasiado, aún utilizando una supercomputadora masivamente paralela.

¿Cuándo se utiliza?

El algoritmo de Diffie-Hellman se suele utilizar en situaciones en las que se necesita un intercambio rápido de claves.

Muchas veces, el software está programado para cambiar los valores de sus claves con cierta frecuencia (a veces, el cambio tiene lugar después de cada transacción), con lo que se necesita un mecanismo rápido que permita la generación de un secreto que sea compartido por ambas partes, incluso aunque sólo se disponga de canales públicos para llevarlo a cabo. De ese modo, dos partes contrarias pueden utilizar Diffie-Hellman para producir un valor de secreto compartido, que puede ser utilizado como clave común para un algoritmo de encriptación por clave secreta.

Ataque al algoritmo Diffie-Hellman



Pero, como irás ya comprendiendo, no existe ningún método perfecto, y Diffie-Hellman no es, por desgracia, la excepción a la regla, y es que hay un problema: cuando Inés recibe la tripleta (47, 3, 28), ¿cómo sabe que es de Pablo y no de un atacante?

No hay manera de saberlo.

Desgraciadamente, un atacante puede explotar este hecho para engañar tanto a Pablo como a Inés, como se ilustra en la imagen.

Aquí, mientras Pablo e Inés seleccionan x e y respectivamente, un atacante podría seleccionar su propio número aleatorio z .

Pablo envía el mensaje 1 destinado a Inés. El atacante lo intercepta y envía el mensaje 2 a Inés, utilizando el g y el n correctos (que de todas maneras son públicos) pero con su propia z en lugar de x . El atacante también envía el mensaje 3 de regreso hacia Pablo.

Después, Inés envía el mensaje 4 a Pablo, que es interceptado y guardado también por el atacante.

Ahora, todos hacen aritmética modular. Pablo calcula la clave secreta como $g^{xz} \bmod n$, y también lo hace el atacante (para los mensajes de Pablo).

Inés calcula $g^{yz} \bmod n$, al igual que el atacante (para los mensajes de Inés).

Pablo piensa que está hablando con Inés, por lo que establece una clave de sesión (con el atacante).

Lo mismo hace Inés. Cada mensaje que Pablo envía durante la sesión cifrada es capturado por el atacante, almacenado, modificado si él así lo desea, y pasado (opcionalmente) a Inés.



Lo mismo ocurre en la otra dirección. El atacante lo ve todo y puede modificar todos los mensajes si lo desea, mientras que tanto Pablo como Inés están pensando, equivocadamente, que tienen un canal seguro entre ambos.

Seguro que te habrás dado cuenta de que este **ataque es el ataque de tercero interpuesto** (the man-in-the-middle attack), aunque también se le conoce por el nombre de **ataque de brigada de cubetas**, porque se parece vagamente a las brigadas de bomberos voluntarios antiguas en las que se pasaban las cubetas de agua en fila desde el carro de bomberos hasta el incendio.



4.7 Algoritmo RSA y Certificados Digitales

Introducción a la Sección 4.7

Vas a comenzar el [apartado 4.7](#):

Definición

Uno de los primeros esquemas de clave pública fue desarrollado en 1977 por Ron Rivest, Adi Shamir y Len Adleman en el MIT; y publicado por primera vez en 1978. El esquema RSA ha sido considerado desde entonces como la única técnica mundialmente aceptada e implementada de algoritmo de encriptado de clave pública.

RSA proporciona un mecanismo de seguridad basado en el hecho de que puede ser relativamente fácil multiplicar números primos entre sí, pero es casi imposible factorizar el resultado de tal producto. Esta técnica produce claves públicas que están ligadas a claves privadas específicas.

Algoritmo RSA paso a paso

Generación de clave	
Seleccionar p, q	p y q ambos primos
Calcular $n = p \times q$	
Seleccionar entero d	$\text{mod } (\Phi(n), d) = 1; 1 < d < \Phi(n)$
Calcular e	$e = d^{-1} \text{ mod } \Phi(n)$
Clave pública	$KU = \{e, n\}$
Clave privada	$KR = \{d, n\}$
Encriptado	
Texto en claro	$M < n$
Texto cifrado	$C = M^e \text{ (mod } n)$
Desencriptado	
Texto cifrado	C
Texto en claro	$M = C^d \text{ (mod } n)$

Es decir, matemáticamente, para algún texto en claro M y un bloque cifrado C, el encriptado y el desencriptado son de la siguiente forma:

$$C = M^e \text{ mod } n$$

$$M = C^d \text{ mod } n = (M^e)^d \text{ mod } n = M^{ed} \text{ mod } n$$

Ambos, el emisor y el receptor deben conocer el valor de n. El emisor conoce el valor de e y el receptor sólo debe conocer el valor de d.

Por tanto, éste es un algoritmo de clave pública con una clave pública dada por $KU = \{e, n\}$ y una clave privada $KR = \{d, n\}$.

Simplificando, el resumen del algoritmo RSA:

Se empieza por seleccionar dos números primos, p y q y calculando su producto n, que es el módulo para el encriptado y el desencriptado.

A continuación, necesitamos la cantidad

$\Phi(n)$, que se conoce como totalizador ("totient") de Euler de n, y que es el número de enteros positivos menores que n y relativamente primos a n.

Entonces, se selecciona el entero d que es relativamente primo a $\Phi(n)$, esto es, el máximo común divisor de d y $\Phi(n)$ debe ser 1.

Finalmente, se calcula e como la inversa del multiplicador de d, módulo $\Phi(n)$.

Se puede demostrar que d y e tienen las propiedades deseadas.

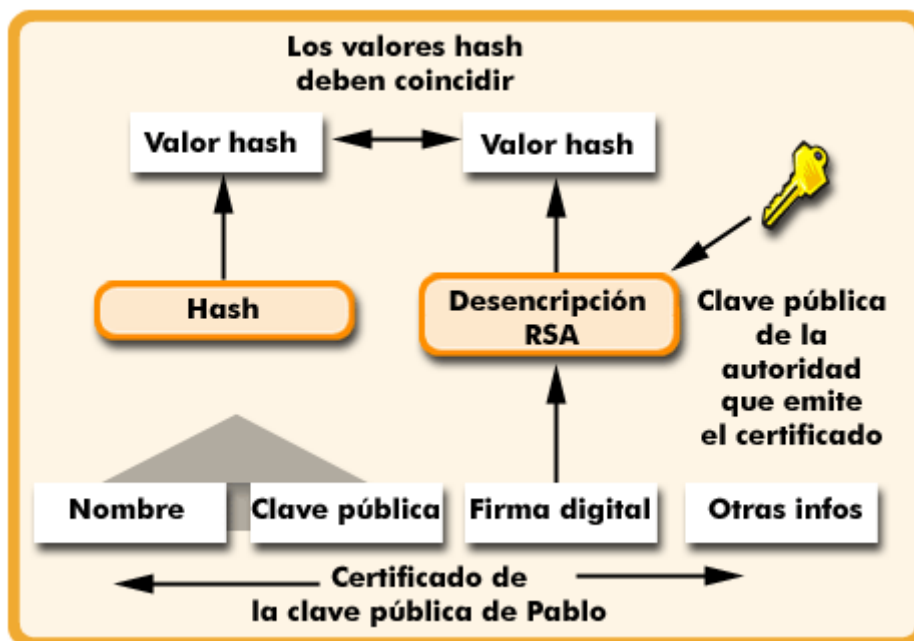
La clave privada resulta ser $\{d, n\}$ y la pública $\{e, n\}$.



Supóngase que el usuario A ha publicado su clave pública y que el usuario B quiere enviar el mensaje M a A.

Entonces, B calcula $C = M^e \pmod{n}$ y transmite C. Cuando se recibe este texto cifrado, el usuario A lo descripta mediante el cálculo $M = C^d \pmod{n}$.

Certificados de Clave Pública



Aunque hemos empleado bastante tiempo en describir cómo puede utilizarse la encriptación y la autenticación y qué papeles juegan las claves públicas y las privadas, hemos dicho muy poco sobre cómo se generan y distribuyen estas claves.

Los servicios de seguridad que posibilitan todo esto se agrupan bajo el término "paraguas" de Infraestructura de Clave Pública (**PKI**, <i>i

Una PKI permite a una organización definir los dominios de seguridad en los que ésta emite claves y sus certificados asociados, que son objetos electrónicos usados para emitir y validar claves públicas.

Una PKI hace posible no sólo la utilización de claves y certificados, sino también la gestión de claves, certificados y políticas de seguridad.

Sin un sistema de este tipo, la utilización de claves públicas sería caótica, ineficiente, inmanejable y, probablemente, insegura.

Los certificados de clave pública están formateados en bloques especiales de datos que nos informan sobre el valor de una clave pública, el nombre del propietario de la clave y la firma digital de la organización que lo emite, llamada "autoridad certificadora" (CA, *Certificate Authority*).

Estos certificados se utilizan para identificar al propietario de una clave pública determinada.

En su forma más simple, un certificado consiste en una clave pública y el nombre de su propietario.

Este certificado es firmado por una CA, cuya clave pública es fácilmente verificable.

Adicionalmente, puede contener la fecha de expedición del certificado, la de expiración de la clave, el nombre del notario electrónico que emitió el certificado y un número de serie.

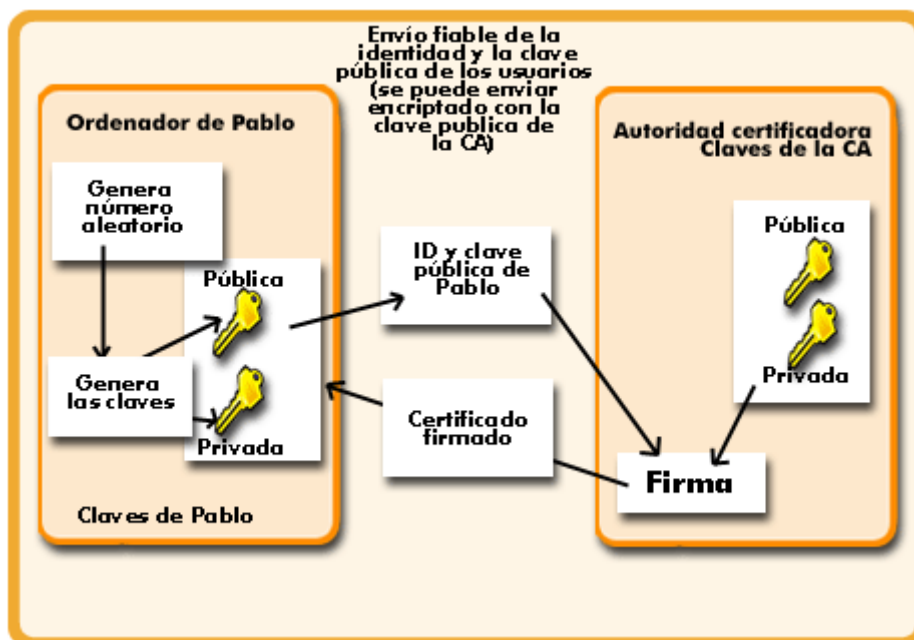


De todo ello, calcula la huella digital con la función de *hash* adecuada y la cifra con su clave privada.

Así, cualquiera que conozca la clave pública de la agencia certificadora correspondiente, podrá ahora verificar que la clave pública de Pablo es auténtica, porque viene firmada por dicha agencia.

Distribución de claves y certificados

Aunque, como ya hemos visto, es menos peligroso distribuir claves públicas que claves privadas, ya que capturando una clave pública no se podrá descifrar ningún mensaje,...



...eso no quiere decir que podamos enviar despreocupadamente claves públicas por canales no protegidos, ya que aunque los atacantes no puedan descifrar mensajes, sí pueden llevar a cabo el ataque, ya mencionado, de "tercero interpuesto" (*the man-in-the-middle attack*).

Alguien que pueda capturar tráfico que contiene una clave pública puede interponerse en la comunicación, introducir su propia clave pública y engañar a dos usuarios que piensan que tienen una comunicación segura y directa, cuando lo que tienen, en realidad, es una comunicación encriptada con un tercero interpuesto que, indirectamente, les pone en contacto entre sí.

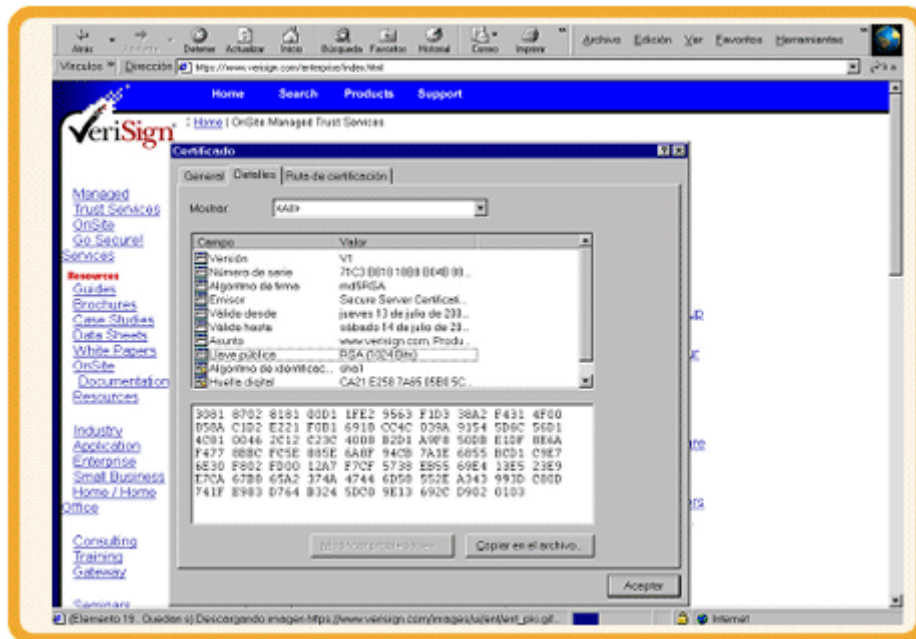
También hemos visto ya cuál es la forma de protegernos frente a este tipo de ataques, la utilización de certificados digitales que garanticen que la clave pública que estoy recibiendo es de quien dice ser.

De ahora en adelante, si Pablo e Inés quieren comunicarse de una forma segura, cada uno de ellos deberá obtener su propio certificado desde una CA de confianza; una vez que tengan su certificado, nunca más volverán a enviar sus claves públicas por el medio de transmisión, sino que se enviarán entre ellos los certificados que atestiguan que las claves públicas que contienen pertenecen a Pablo y a Inés.

Autoridades de certificación

¿Has tenido alguna vez que visitar a un notario?

Bueno, pues también existen notarios electrónicos: las autoridades de certificación, que, como todos los notarios, deben ser entes fiables y ampliamente reconocidos que firman (con conocimiento de causa y asunción de responsabilidades legales) las claves públicas de las personas, rubricando con su propia firma la identidad del usuario.



Debido a la posición comprometida que ocupan las autoridades de certificación, éstas deben tomar extremadas precauciones para evitar que sus claves caigan en manos de intrusos, lo cual comprometería todo el sistema.

Para ello tendrá que utilizar claves largas y dispositivos especiales para su almacenamiento.

Además, cuando emiten un certificado, deben estar seguros de que lo hacen a la persona adecuada.

No podemos olvidar que la autoridad de certificación es la responsable, en última instancia, de todo el proceso, con una serie de responsabilidades legales y que basa su "negocio" en la credibilidad que inspire en sus potenciales clientes.

Una autoridad de certificación con autentificaciones erróneas no tendrá más remedio que cerrar ya que los usuarios no considerarán sus certificados de la suficiente "calidad".

¿Y cómo mantienen su credibilidad? Porque, dado que la clave pública de la Entidad de Certificación es conocida por todos los interlocutores, cualquiera es capaz de extraer los datos del certificado.

Sin embargo, nadie es capaz de suplantar a la Entidad de Certificación emitiendo certificados falsos, ya que carece de su clave privada.

Resumiendo, el proceso es el siguiente:

La Entidad de Certificación (como, por ejemplo, FESTE) comunicará su clave pública a través de los periódicos u otros medios no electrónicos, y proporcionará pruebas de que es una entidad de confianza, por lo que sus certificados pueden ser considerados válidos.

Cuando un participante comunica a otro su certificado, indica la Entidad de Certificación utilizada. La clave pública de la Entidad de Certificación debe ser conocida por todos y es la única que necesita ser conocida previamente.

Habitualmente está incorporada al software de realización y verificación de firmas electrónicas, o es posible obtenerla a partir de sistemas de difusión públicos, tales como servidores Web o Directorios LDAP o X500.

En el caso de FESTE se está haciendo un esfuerzo para que tanto Microsoft como Netscape incluyan las claves públicas de la Entidad de Certificación en su software, al objeto de facilitar su uso por parte de los usuarios españoles.

Es posible que diferentes Entidades de Certificación se relacionen mediante sistemas jerárquicos de forma que el software utilizado sólo necesita integrar la clave pública de la Entidad de Certificación de nivel más alto, y todas las demás Entidades de Certificación de la jerarquía quedan automáticamente validadas.

En este sentido se están desarrollando los estándares PKIX y SPKI.

Sin embargo, los sistemas jerárquicos, viables desde el punto de vista técnico, plantean problemas de subordinación de soberanía que pueden llegar a hacerlos inaceptables en algunos países.



Bienvenido al capítulo:

Técnicas de Autenticación



5.1 Técnicas de Autenticación

Introducción a la Sección 5.1

Vas a comenzar el [apartado 5.1](#):

Definición



¿Nunca te has preguntado cuáles son los porteros de Internet?



¿No has oído hablar de la **autenticación**?

[Ver su definición](#)

La autenticación es una parte vital de la estructura de seguridad de cualquier sistema. A menos que un sistema puede autenticar, de una manera fiable, tanto a los usuarios, como a los servicios y a las redes, no se podrá controlar el acceso a los recursos de la red y mantener alejados de nuestras redes a los usuarios no autorizados.

¿Y qué herramientas se utilizan para comprobar quiénes somos?

[Ver las herramientas](#)

La autenticación está basada en uno de los tres atributos siguientes:

- Algo que tenemos (una clave, una tarjeta de muestra).
- Algo que sabemos (una password).
- Algo que somos (huella vocal, escáner de retina).

Tipos



Por lo general, es comúnmente aceptado entre los expertos en seguridad, que los métodos simples de autenticación, que se basan en sólo uno de los atributos anteriormente mencionados (por ejemplo, los basados sólo en una password), no son adecuados para la protección de sistemas.

En su lugar, los expertos recomiendan lo que se conoce como “autenticación fuerte”, es decir, aquella que utiliza al menos dos de los atributos anteriormente mencionados.

Los sistemas de autenticación habitualmente utilizados pueden clasificarse en diversos tipos.

- Basados en password tradicionales.
- Passwords de una sola utilización (S/Key).
- Otros sistemas de password (PAP, CHAP, TACACS y RADIUS).
- Basados en hardware (tokens, tarjetas inteligentes – smart cards – y tarjetas de PC).
- Basados en identidades biométricas (huella digital, sonido de la voz, escáner de retina).



Autenticación con passwords tradicionales

Todo el mundo reconoce que la forma más simple de autenticación (es decir, utilizar identidades de usuario y passwords) no es adecuada para un acceso seguro a redes.

Las passwords pueden ser adivinadas e interceptadas durante las transmisiones en la red.



Incluso aunque los usuarios sean cuidadosos al guardar los secretos de sus passwords, no se dan cuenta de que los servicios que ofrece Internet no son capaces de proteger sus passwords. Por ejemplo, servicios tales como FTP y Telnet transmiten identidades de usuario y passwords como texto “en claro”, facilitando así su interceptación.

Bueno, pues alguna solución se nos tendrá que ofrecer, ¿no?

Ver solución

Los **sistemas de passwords de un sólo uso**, que restringen la validez de una password a una única sesión, pueden ser una buena solución a algunos de los problemas que conlleva la utilización passwords tradicionales.

5.2 Passwords de un solo uso

Introducción a la Sección 5.2

Vas a comenzar el apartado 5.2:

Definición. Descripción de S/Key



¿Qué puede ser este mecanismo de seguridad?

Pues algo tan simple como su propio nombre indica, ya verás...

Una forma de prevenir el uso no autorizado de una password interceptada es evitar que dicha password pueda ser reutilizada (por ejemplo, restringiendo el uso de las passwords a una única sesión de comunicación).

Como su propio nombre indica, los sistemas que utilizan passwords de un solo uso requieren una nueva password para cada nueva sesión.

Estos sistemas, de los cuales S/Key (desarrollado inicialmente por Bellcore) es el mejor ejemplo, mitigan la dificultad que supone al usuario la utilización de una nueva password para cada sesión, ya que son capaces de generar automáticamente una lista de passwords aceptables para el usuario.

El IETF se ha ocupado de la estandarización de S/Key en las especificaciones para Sistemas OTP (*One Time Passwords*) en RFC 2289.

S/Key utiliza una "frase de paso" secreta (de 10 caracteres, por lo menos) creada por el propio usuario, para generar una secuencia de passwords de un solo uso.

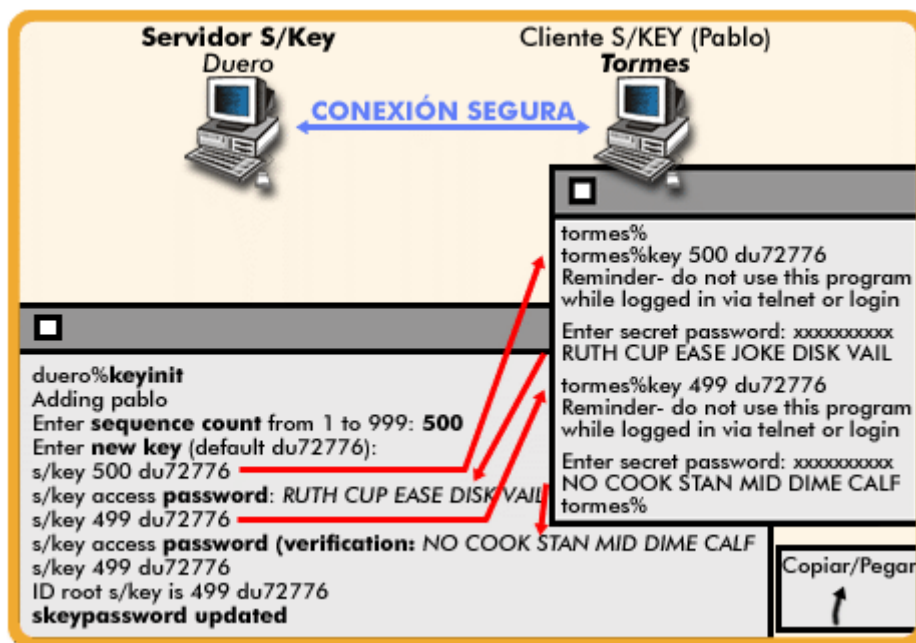
La frase de paso secreta del usuario nunca viaja más allá de su ordenador local y no circula por la red; por lo tanto, la frase de paso no es susceptible de ser atacada.



Además, como se genera una password diferente para cada sesión, una password interceptada no puede utilizarse de nuevo ni tampoco le da información al atacante sobre cuál va a ser la siguiente password que se va a utilizar.

La secuencia de passwords de un solo uso se genera por medio de una función hash (también llamada “función de mezcla”) segura, que se aplica varias veces a la concatenación del “desafío” lanzado por el servidor (la “semilla”) con la frase de paso secreta del usuario.

Un ejemplo de S/Key



Para entender bien el proceso, ¿qué te parece si vemos un ejemplo práctico de utilización de la autenticación S/Key por un usuario llamado Pablo, que se encuentra en un ordenador cliente Tormes y quiere conectar con un ordenador servidor Duero? ¿Cómo sería el inicio del proceso?

Cuando un usuario necesita utilizar el protocolo S/Key de autenticación, debe disponer de un enlace directo local con el ordenador servidor S/Key, nunca a través de telnet o rlogin.

Mediante este enlace instruye al servidor con los parámetros necesarios para que este ordenador pueda validar las secuencias de passwords que el usuario utilizará remotamente en sus conexiones remotas posteriores.

Una vez que estamos conectados directamente a la máquina Duero, podemos iniciar el proceso de autenticación S/Key por medio de la ejecución del comando **keyinit**.

La ejecución del comando Keyinit provocará que Duero nos pida un número de secuencia que consistirá en el número de conexiones que podemos establecer con dicha máquina, y que se irá decrementando cada vez que establezcamos una nueva conexión (por tanto, se recomienda elegir un número elevado porque, de otro modo, rápidamente nos quedaremos sin poder establecer nuevas conexiones remotas, y puede que no estemos en las cercanías de Duero para volver a iniciar el proceso).

También nos pedirá Duero una "semilla" en forma de caracteres alfanuméricos; podemos escoger la que se nos propone por defecto.

Tanto el número de secuencia y la semilla formarán parte del "desafío" que el servidor lanzará al cliente cada vez que éste quiera establecer una conexión (vía telnet o rlogin) con aquel.

Ahora hay que introducir en Duero la password correspondiente a la iteración 500 que, evidentemente, deberá ser calculada por el ordenador Tormes, ya que dicha password se genera al iterar 500 veces una función hash (MD4 o MD5) sobre la concatenación de la "semilla" con la frase secreta de paso, y esta última no se envía nunca a Duero.

Por tanto, invocamos al comando **key 500 du7276** para que Tormes calcule la password correspondiente a la iteración 500. El resultado lo "pegamos" en la ventana de la conexión con Duero, y así éste aprende cuál es la password con la que va a validar la primera conexión de Pablo desde Tormes.

Pero, antes de que Duero considere válida esta password, pide una verificación. La verificación consiste en que nos pide la password correspondiente a la iteración 499 (siempre con la misma semilla).

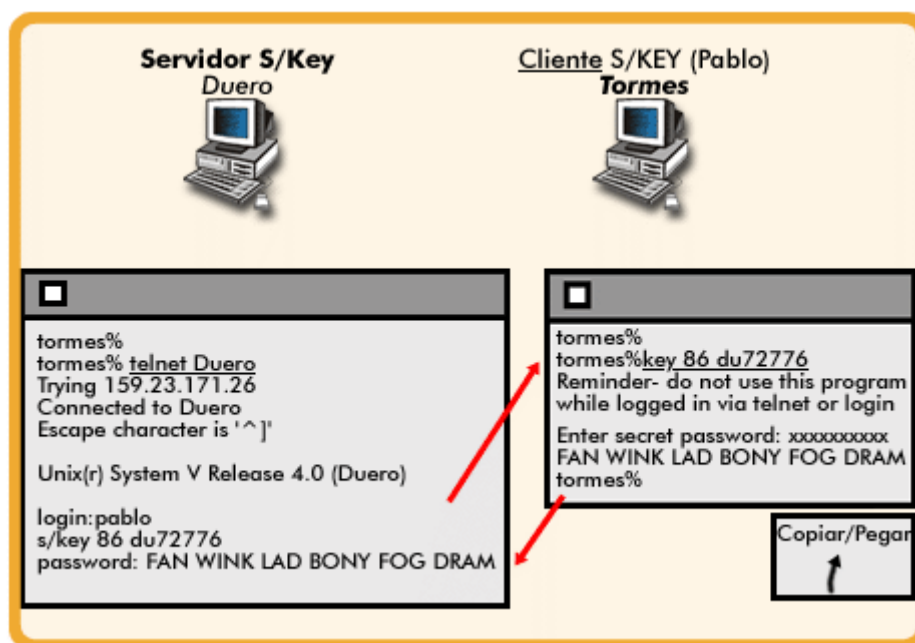
Esta nueva password la volvemos a calcular en Tormes, porque es el único que conoce la frase de paso secreta, por lo que volvemos a invocar al comando **key 499 du7276** para que Tormes calcule la password correspondiente a la iteración 499. El resultado, lo "pegamos" en la ventana de la conexión con Duero, y así éste inicia el proceso de verificación.

Aplica la función hash (MD4 o MD5) a la password enviada desde Tormes; es decir, realiza una nueva iteración a la iteración número 499.

El resultado debería coincidir con la password que Duero tiene almacenada de la iteración 500. Si es así, confirma la validez.

Una vez confirmada la password correspondiente a la iteración 500, se da por finalizada la inicialización. Ahora, Pablo podrá establecer 500 conexiones remotas (vía telnet o rlogin) pero utilizando S/Key con passwords de un solo uso.

Un ejemplo de S/Key (II)



Una vez inicializado el proceso, vamos a seguir con el desarrollo de una conexión por telnet desde Tormes a Duero en la cuenta que Pablo posee en esta última máquina.

Los sistemas de passwords de un solo uso, como S/Key, requieren que el software del servidor sea modificado para realizar los cálculos requeridos, y que cada máquina remota tenga una copia del software cliente.

Estos sistemas no suelen ser muy escalables, debido a la dificultad que entraña la administración de listas de passwords para grandes cantidades de usuarios.

Cuando Pablo, desde la máquina Tormes, quiera acceder, mediante telnet, a la máquina Duero, procederá invocando telnet Duero. Cuando introduzca su login "pablo", Duero procederá a ejecutar el mecanismo de autenticación S/Key, para lo cual desafiará a Pablo a que le introduzca la password correspondiente a la iteración 86 (se supone que Pablo ya ha "consumido" 414 conexiones).

Pablo copiará el desafío en una ventana de Tormes que no esté bajo conexión telnet y forzará 86 ejecuciones de la función hash, concatenando la semilla contenida en el desafío, du72776, con la frase de paso secreta que habrá que introducir de nuevo.

El resultado de esta ejecución lo copiará Pablo en la ventana telnet para que Duero compruebe si ejecutando una iteración más (la 87) con la password que ha generado Tormes, se obtiene la password que Duero tenía almacenada desde la última conexión.

Si es así, Duero guardará la password correspondiente a la iteración 86 (para validar la próxima conexión) y permitirá el acceso de Pablo.



5.3 Otros sistemas de autenticación

Introducción a la Sección 5.3

Vas a comenzar el [apartado 5.3](#):

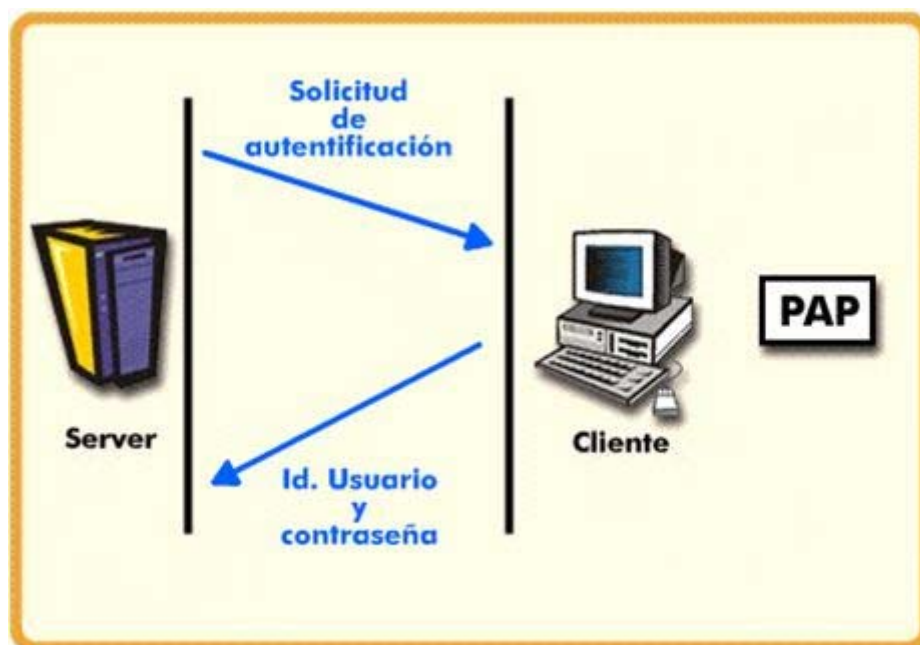
Introducción

Me imagino que te preguntarás si pueden existir más alternativas a los procedimientos de autenticación que ya hemos visto, y tenemos la fortuna de que así ocurre en la realidad.

Más allá de los métodos tradicionales de autenticación por passwords, que a menudo requieren del envío “en claro” (no encriptado) tanto de la identidad de usuario como de la propia password, existen otros mecanismos de autenticación basados en passwords que se han desarrollado especialmente para el acceso remoto.

Estos métodos son los utilizados habitualmente por los sistemas de Redes Privadas Virtuales (VPN) y aquí estudiaremos algunos de ellos: PAP, CHAP, TACACS y RADIUS.

Password Authentication Protocol (PAP)



Pero veámoslos por orden, que no tenemos prisa...

El protocolo de autenticación por password (PAP) fue diseñado en un principio como una forma simple de que un ordenador se autentificase a sí mismo frente a otro cuando se utiliza PPP como protocolo de comunicación.

PAP es un protocolo de handshaking de dos vías; es decir, el host que establece la conexión envía el par identidad de usuario / password, al sistema con el que está intentando establecer la conexión, y dicho sistema (el autenticador) reconoce al ordenador que está intentando la autenticación y aprueba o no la comunicación.

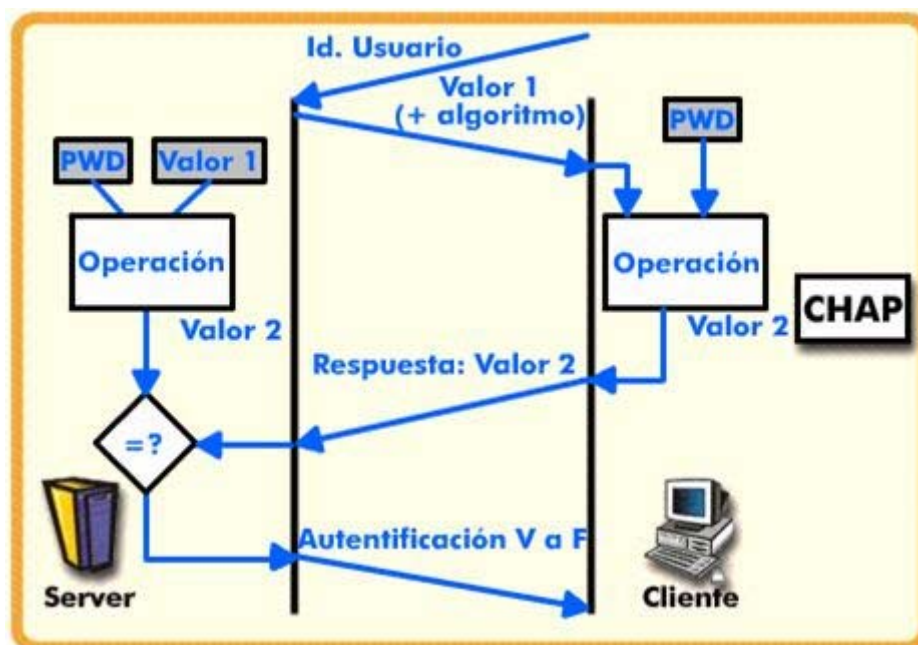
La autenticación PPP puede utilizarse en el establecimiento del enlace PPP así como durante la sesión PPP para volver a autenticar el enlace.

Cuando se establece el enlace PPP, la autenticación PAP se realiza sobre dicho enlace.

Uno de los extremos emite la identidad de usuario y la password "en claro" hacia el autenticador, hasta que el autenticador las acepte o se cierre la conexión.

PAP no es seguro porque la información de autenticación se transmite "en claro" (no encriptada), con lo que no hay protección frente a los ataques "por repetición" de la password previamente capturada por un atacante, ni de los ataques por "excesivas repeticiones" tratando de adivinar el par válido identidad de usuario / password ("ataques de diccionario", por ejemplo).

Challenge Handshake Authentication Protocol (CHAP)



CHAP fue diseñado para los mismos propósitos que PAP, aunque se trata de un método más seguro de autenticación de enlaces PPP. CHAP es un protocolo de handshaking de tres vías.

Al igual que PAP, CHAP puede utilizarse durante el establecimiento de la conexión PPP y repetido con posterioridad para reautenticar el enlace.

¿Qué es eso de handshaking de tres vías? Decimos que CHAP es un protocolo de handshake de tres vías porque necesita tres pasos para autenticar el enlace.

En vez de utilizar un proceso simple de autenticación basada en password de dos etapas, como hacía PAP, CHAP utiliza una función de mezcla (hash) de forma similar a como lo hacía S/Key. El proceso tiene lugar como se explica a continuación.

El autenticador genera un valor aleatorio (que será diferente en cada petición de autenticación) y lo envía como mensaje de desafío a su par.

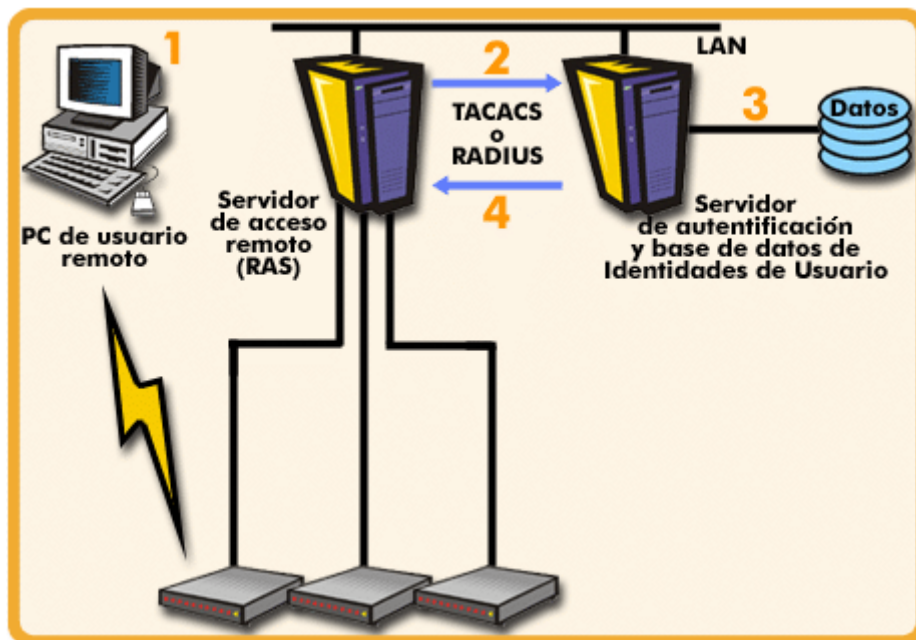
Al mismo tiempo, utiliza una función hash de una vía para calcular un nuevo valor, suministrándole como entrada a la función la concatenación del valor aleatorio con la password del usuario que pretende autenticarse.

El par utiliza la misma función hash con las mismas entradas y calcula un valor para devolverlo al autenticador.

El autenticador puede aceptar la autenticación si el valor concuerda con el que él mismo ha calculado.

Ventajas de CHAP

¿Qué ventajas proporciona CHAP?



Para asegurar que la conexión no ha sido “robada” o alterada de alguna forma, el proceso se puede repetir tantas veces como se quiera durante el tiempo que dure el enlace PPP.

Al contrario que PAP, que es controlado desde el host del cliente, es el servidor el que controla la reautenticación CHAP.

CHAP elimina la posibilidad, inherente a PAP, de que un atacante pueda efectuar intentos repetidos de entrar en la misma conexión.

Cuando la autenticación CHAP falla, el servidor dará por finalizada la conexión.

Esto dificulta la labor del atacante de adivinar la password, porque no puede intentar ataques repetidos en la misma conexión.



Desventajas de PAP y CHAP

Ambos mecanismos se basan en la utilización de passwords secretas que deben ser almacenadas tanto en ordenadores remotos como en los ordenadores locales de los usuarios. Si cualquiera de estos ordenadores sucumbe al ataque de un "pirata", el secreto de las passwords se ve comprometido.

Además, ni PAP ni CHAP permiten la asignación de diferentes privilegios a usuarios remotos que acceden desde el mismo ordenador, ya que los privilegios se asignan por ordenador remoto, y no por usuario. Los dos protocolos que vamos a presentar a continuación, TACACS y RADIUS, proporcionan mayor flexibilidad al asignar privilegios de acceso.

Aunque CHAP es un método más robusto que PAP en la autenticación de usuarios "por marcación", CHAP no satisface los requisitos de escalabilidad de grandes organizaciones.

A pesar de que los secretos no se transmiten por la red, requiere que un gran número de secretos sean compartidos para que puedan ser ejecutados por las funciones hash de cada ordenador.

Las organizaciones que poseen un gran número de usuarios que acceden "por marcación" deben mantener grandes bases de datos para acomodarlos.



Terminal Access Controller Access-Control System (TACACS)

TACACS es uno de los sistemas desarrollados para proporcionar no sólo autenticación, sino las otras dos As de la triple A (AAA, Authentication, Authorization and Accounting).

Al contrario que PAP y CHAP, que funcionan con un modelo de relaciones entre pares, tanto TACACS como RADIUS (la otra técnica que veremos a continuación) fueron diseñados para funcionar según el modelo cliente/servidor, que ofrece una mayor flexibilidad, especialmente en la gestión de la seguridad. El elemento central de operación de TACACS y RADIUS es el servidor de autenticación.

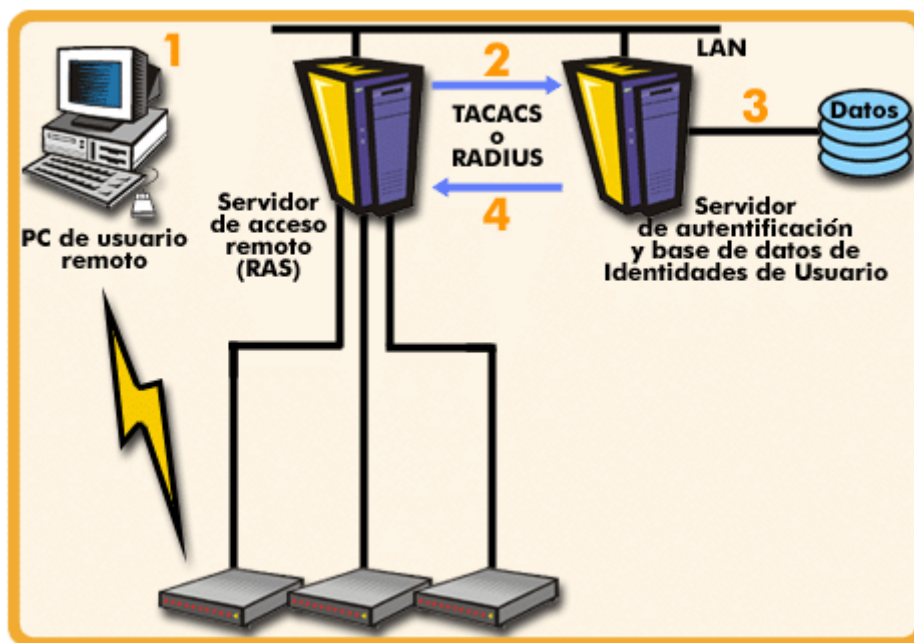
Pero, centrándonos en TACACS, ¿cómo funciona concretamente?

Ver cómo funciona

Generalmente, un servidor de autenticación TACACS gestiona peticiones de autenticación provenientes de un software cliente que se encuentra situado en un gateway o punto de entrada a la red.

El servidor de autenticación mantiene una base de datos de identidades de usuarios, passwords, PINs y claves secretas, que utiliza para conceder o denegar peticiones de acceso a la red. Todos los datos de autenticación, autorización y tarificación se envían hacia el servidor central cada vez que un usuario intenta acceder a la red.

Ventajas e inconvenientes de TACACS



Luego veremos las ventajas e inconvenientes.

- 1) El usuario marca al servidor de acceso remoto (RAS).
- 2) Utilizando los protocolos TACACS o RADIUS, el RAS envía peticiones de autenticación/autorización al servidor de autenticación.
- 3) El servidor de autenticación chequea las peticiones contra las identidades de la base de datos.
- 4) Vía TACACS o RADIUS, el servidor de autenticación envía instrucciones al RAS para conceder o denegar el acceso.



Ventajas

Una ventaja de TACACS es que puede actuar como servidor proxy para otros sistemas de autenticación, tales como dominios de seguridad de Windows NT, NDS, mapas NIS basados en Unix, y otros sistemas de seguridad (tales como los sistemas basados en tokens, que mencionaremos brevemente más adelante).

Las capacidades proxy facilitan que un cliente de una corporación comparta los datos en una VPN segura cuando dicha VPN es subcontratada a un ISP; el ISP implementa un servidor proxy para controlar el acceso de los clientes por marcación basándose en los derechos de acceso del cliente corporativo que son gestionados por el propio servidor corporativo.



Inconvenientes

El principal inconveniente de TACACS es que transmite todos los datos sin encriptar entre el servidor y el cliente, pero Cisco ha incorporado una actualización reciente, TACACS+, que añade una función de resumen a los mensajes para eliminar la transmisión de passwords sin encriptar. TACACS+ también soporta logins multiprotocolo; es decir, una misma identidad de usuario puede utilizarse para autenticar a un usuario en múltiples equipos y redes (por ejemplo, un login a una red IP y otro a una red IPX). Finalmente, TACACS+ también maneja autenticación PAP y CHAP.

La transmisión de los paquetes de autenticación entre el servidor corporativo y el servidor proxy a través de una red pública representa un riesgo a la seguridad. La encriptación RADIUS y TACACS está basada en claves estáticas; los nombres de usuario, las passwords y la información del servidor de autenticación están contenidas en un paquete único, lo que facilita su utilización si dicho paquete es interceptado.

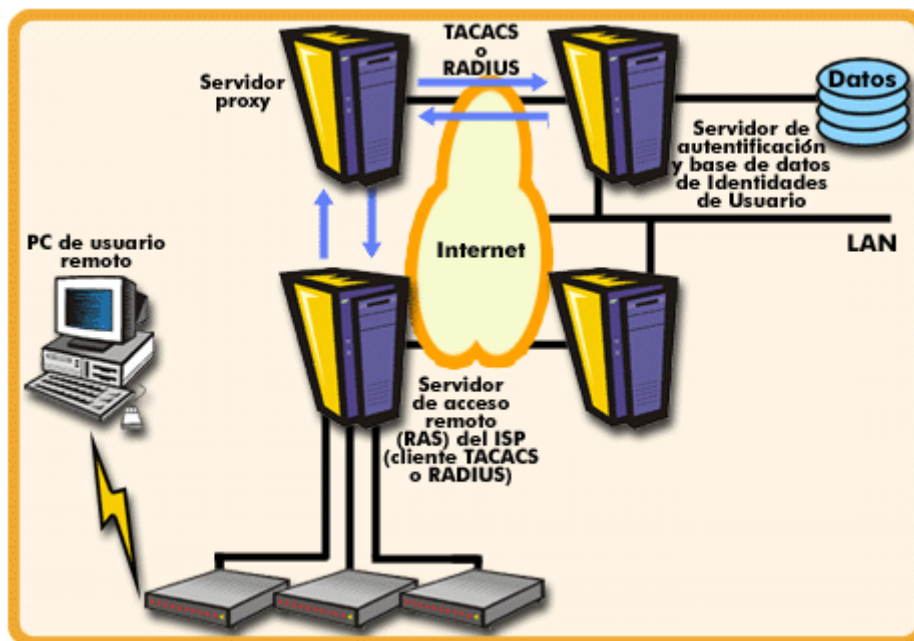
Y, además, TACACS ofrece un inconveniente adicional, como es el rechazo que tiene entre algunos fabricantes. ¿Por qué?

TACACS es conocido como el protocolo software de seguridad basado en servidores de los Sistemas Cisco. Todas las familias de routers y de servidores de acceso de Cisco utilizan este protocolo. Aunque TACACS está descrito en una RFC del IETF, y es de libre utilización por parte de otros fabricantes, muchos de ellos consideran a TACACS como una solución propietaria y centran sus esfuerzos en RADIUS.

Remote Authentication Dial-In User Service (RADIUS)

¿Y cuál es la alternativa a TACACS?

Pues el protocolo RADIUS.



El protocolo RADIUS también utiliza el modelo cliente/servidor para la autenticación segura y la administración de sesiones y conexiones de usuario a redes remotas.

Hoy día, RADIUS se presenta como una herramienta, ampliamente difundida, para realizar el control del acceso de una forma manejable; además, soporta otros tipos de autenticación, incluyendo PAP y CHAP.

El modelo cliente/servidor de RADIUS utiliza un NAS (Network Access Server) para gestionar las conexiones de usuarios.

Aunque el NAS funciona como un servidor para proporcionar el acceso a redes, también funciona como cliente RADIUS.

El NAS es responsable de aceptar las peticiones de conexión de los usuarios, de obtener la información de identidad de usuario y password, y de hacérsela llegar, de forma segura, al servidor RADIUS.

El servidor RADIUS devuelve la concesión o la negación de la autenticación así como cualquier dato de configuración requerido por el NAS para proporcionar servicios al usuario final.

Los clientes y los servidores RADIUS se comunican de forma segura mediante secretos compartidos para autenticación y encriptación.

RADIUS crea una base de datos de usuarios y de servicios disponibles, única y centralizada.



Ésta es una característica particularmente importante para redes que incluyen grandes bancos de módems, y más de un servidor de comunicaciones remotas (más de un NAS).

Con RADIUS, la información de usuario se mantiene en una localización, el servidor RADIUS, que gestiona la autenticación del usuario y el acceso a los servicios desde dicha localización.

Como cualquier equipo que soporte RADIUS puede trabajar como cliente RADIUS, un usuario remoto tiene la posibilidad de poder acceder a los mismos servicios desde cualquier servidor de comunicaciones (NAS) que se comunique con el servidor RADIUS.



Bienvenido al capítulo:

Protocolos de seguridad

6.1 Protocolos de seguridad

Introducción a la Sección 6.1

Vas a comenzar el apartado 6.1:

Introducción

PROTOCOLOS DE SEGURIDAD: SSL – PGP – SET - IPsec

¿Qué entendemos como protocolo de seguridad?

Un protocolo de seguridad es la parte visible de una aplicación; es el conjunto de programas y actividades programadas que cumplen con un objetivo específico y que usan esquemas de seguridad criptográfica.

- PRINCIPALES PROTOCOLOS



Evidentemente, no existe un único protocolo de seguridad, sino que nos encontramos con varios protocolos distintos. Los **principales protocolos** son los siguientes:

El protocolo más común es **SSL** (Secure Sockets Layer) (que vemos integrado en el Browser de Netscape y hace su aparición cuando el candado de la barra de herramientas se cierra y también si la dirección de internet cambia de http a https); otro existente es **PGP** que es un protocolo libre, ampliamente usado, de intercambio de correo electrónico seguro; uno de los más conocidos es el muy publicitado **SET** que es un protocolo que permite dar seguridad en las transacciones por internet usando tarjeta de crédito. Por último, nombraremos a **IPsec**, que proporciona seguridad en la conexión de internet a un nivel más bajo.

- FUNCIÓN DE UN PROTOCOLO



Y, básicamente, la **función de un protocolo** es la de integrar y resolver algunos de los problemas de seguridad que hemos visto en capítulos anteriores, como la integridad, la confidencialidad, la autenticación y el no rechazo, mediante sus diferentes características. Las características de los protocolos se derivan de las múltiples posibilidades con que se puede romper un sistema, es decir, robar información, cambiar información, leer información no autorizada, y todo lo que se considere no autorizado por los usuarios de una comunicación por red.

Veámoslos por parte y en orden... ¿Empezamos con SSL?

6.2 Protocolo SSL

Introducción a la Sección 6.2

Vas a comenzar el apartado 6.2:

Definición y características



Secure Sockets Layer (SSL) es un protocolo diseñado por Netscape Communications Co., que proporciona un nivel seguro de transporte entre el servicio clásico de transporte en Internet (TCP) y las aplicaciones que se comunican a través de él.

SSL

Es un protocolo que proporciona conexiones seguras sobre una red insegura como es Internet, garantizando:

- **Conexión privada:** la información se cifra utilizando criptografía de clave simétrica.
- **Autenticación:** usando criptografía de clave pública.
- **Integridad:** la integridad de los mensajes se asegura usando firmas digitales.

Handshake SSL

El protocolo SSL, se compone de dos partes diferenciadas, siendo una de ellas **Handshake**, la cual se encarga de establecer la conexión y determinar los parámetros que se van a utilizar posteriormente.

(Fundamentalmente se trata de establecer cuál va a ser la clave simétrica que se utilizará para transmitir los datos durante esa conexión).

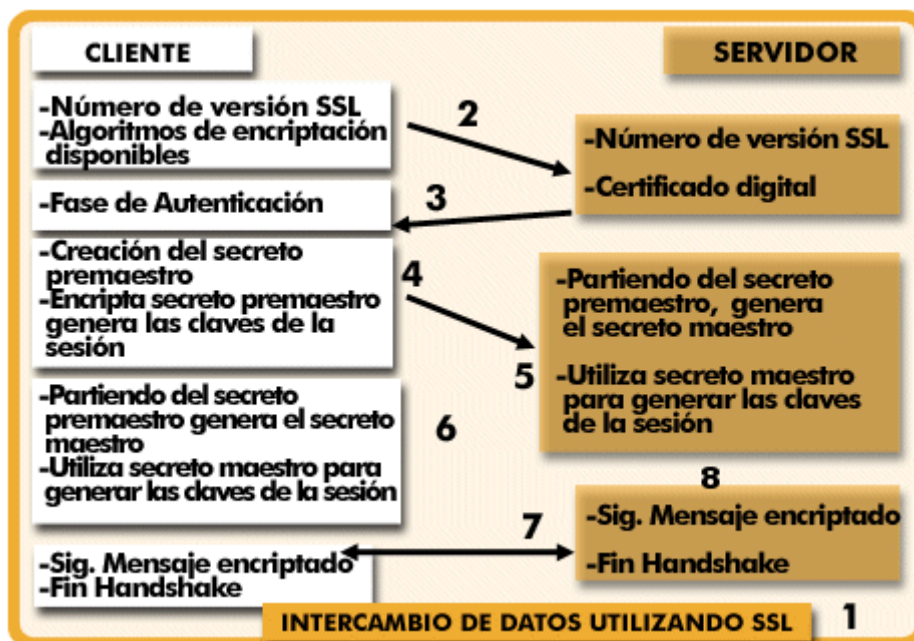
Registro SSL

La otra de las partes diferenciadas del protocolo SSL es el **registro**, el cual comprime, cifra, descifra y verifica la información que se transmite.

Y la gran ventaja de todo ello es que este sistema es *transparente para las aplicaciones finales*, que, simplemente, saben que el canal se encarga de proporcionarles confidencialidad entre extremos.

Por tanto, podemos situar protocolos como HTTP, FTP, NNTP o Telnet.

Handshake SSL



Internamente, el protocolo SSL utiliza una combinación de encriptación de clave pública y simétrica, ya que aunque la encriptación simétrica es mucho más rápida que la de clave pública, la de clave pública proporciona técnicas de autenticación mucho más seguras.

1

Veamos el proceso desde el principio: una sesión SSL siempre comienza con un intercambio de mensajes denominado SSL handshake, que permite al servidor autenticarse frente al cliente, utilizando técnicas de clave pública (RSA, generalmente), permitiendo acto seguido la cooperación entre el cliente y el servidor para la creación de las claves simétricas que se utilizarán para una rápida encriptación, desencriptación y detección de la desnaturalización de los datos durante la sesión que comienza inmediatamente.

Opcionalmente, el handshake también permite que el cliente se autentique frente al servidor.

2

De forma más concreta, en el paso inicial de dicho proceso, el cliente envía al servidor el número de versión del SSL de que dispone, los algoritmos de encriptación que puede utilizar, datos generados aleatoriamente y más información que el servidor necesita para comunicarse con el cliente mediante SSL.

3

El servidor envía al cliente los datos anteriores (versión de SSL de que dispone el servidor, etc.) al cliente. El servidor envía también su propio certificado (incluyendo la clave pública) y, si el cliente está pidiendo un recurso del servidor que requiere autenticación del cliente, pide el certificado del cliente.

El cliente utiliza parte de la información enviada por el servidor para autenticar al servidor, como veremos más adelante. Si el servidor no puede ser autenticado, el usuario es avisado de la existencia de un problema y se le informa de que no puede establecerse la conexión encriptada y autenticada. Si el servidor es autenticado con éxito, continuamos.

4

El cliente (con la cooperación del servidor, dependiendo del algoritmo de encriptación que haya sido seleccionado por éste) crea el **secreto premaestro** (premaster secret) para la sesión, lo encripta con la clave pública del servidor (obtenida del certificado del servidor), y envía el secreto premaestro al servidor.

Si el servidor pidió autenticación al cliente, el cliente también firma otra pieza de datos que va a ser única para este handshake y conocida por ambos, el cliente y el servidor. En este caso, el cliente envía los datos firmados y su propio certificado al servidor junto con el secreto premaestro encriptado.

5

Si el servidor pidió autenticación del cliente, el servidor intenta autenticar al cliente, con los métodos que veremos más adelante. Si el cliente no puede ser autenticado, se finaliza la sesión.

Por el contrario, si es autenticado con éxito, el servidor utiliza su propia clave privada para desencriptar el secreto premaestro, y realiza después una serie de operaciones (que también realiza el cliente al mismo tiempo) partiendo del mismo secreto premaestro para determinar el **secreto maestro**.

Al final, ambos, cliente y servidor obtienen el mismo secreto maestro.

6

Los dos, cliente y servidor, utilizan el secreto maestro para generar las claves de la sesión, que son las claves simétricas que se van a utilizar para encriptar y desencriptar la información durante la sesión SSL, y para verificar su integridad (es decir, para detectar cambios en los datos mientras circulaban por la red).

7

El cliente envía un mensaje al servidor informándole de que los mensajes futuros desde el cliente se van a encriptar con la clave de la sesión.

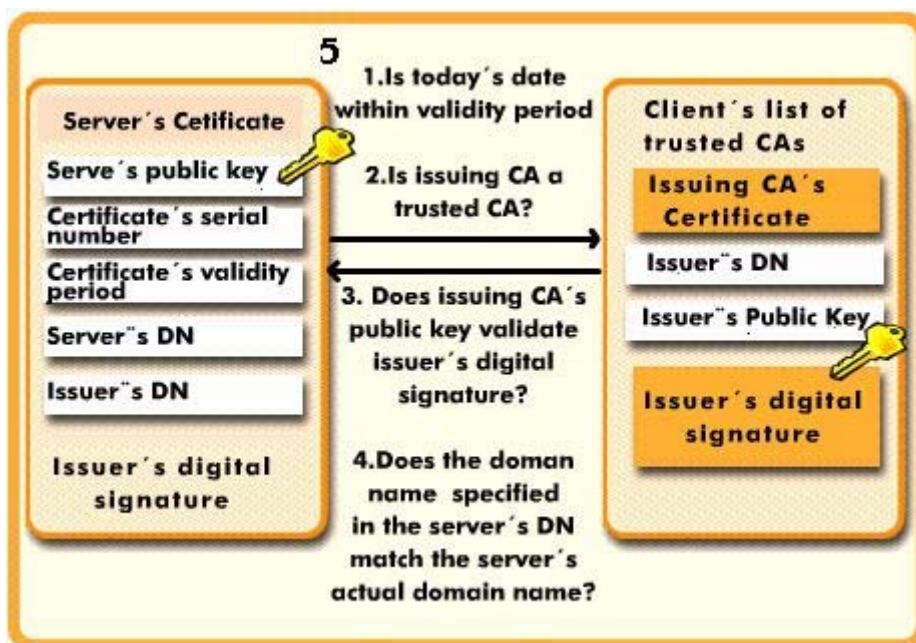
Después envía un mensaje separado (encriptado), indicando que la parte de handshake del cliente ha finalizado.

8

El servidor envía un mensaje al cliente informándole de que los mensajes futuros desde el servidor se van a encriptar con la clave de la sesión. Después envía un mensaje separado (encriptado) indicando que la parte de handshake del servidor ha finalizado.

El handshake SSL se ha completado y ha comenzado la sesión SSL. El cliente y el servidor utilizan las claves de la sesión para encriptar y desencriptar los datos que se envían entre ellos y para validar su integridad.

Autenticación del servidor



Vimos que los clientes SSL siempre piden autenticación al servidor, es decir, validación criptográfica de la identidad del servidor por el cliente. Para ello, el servidor envía al cliente el certificado que le autentica a él mismo. ¿Y cómo se realiza la autenticación? Pues preguntando. Para autenticar la relación entre una clave pública y el servidor identificado por el certificado que contiene la clave pública, un cliente SSL debe recibir una respuesta afirmativa a cuatro preguntas.

¿Está la fecha de hoy dentro del período de validez del certificado?

El cliente comprueba el período de validez del certificado del servidor.

Si el día y la hora actual están fuera de dicho rango, el proceso de autenticación no seguirá más allá.

Por el contrario, si lo están, seguimos el proceso.

¿Es la autoridad certificadora (CA) una CA de confianza?

Cada cliente SSL mantiene una lista de CAs de confianza. Esta lista determina los certificados de servidores que el cliente aceptará.

Si el nombre distinguido (distinguished name, DN) de la entidad CA emisora concuerda con algún CA de los de la lista del cliente de CAs de confianza, la respuesta a esta pregunta es "sí", y el cliente continúa el proceso. Si, por el contrario, el DN del CA emisor no está en la lista, el servidor no será autenticado, a menos que el cliente pueda verificar un certificado de un CA que no está en su lista (por procedimientos de jerarquías de CA).

¿La CA que emitió la el certificado (con la clave pública) valida la firma digital del emisor de dicho certificado?

El cliente utiliza la clave pública del certificado del CA para validar la firma digital del CA presentada en el certificado del servidor. Si la información en el certificado



del servidor ha cambiado desde que fue firmada por el CA, o si la clave pública del certificado del CA no corresponde con la clave privada utilizada por el CA para firmar el certificado, el cliente no podrá autenticar la identidad del servidor. Si la firma digital del CA se puede validar, el cliente trata el certificado del servidor como una “carta de presentación” válida de ese CA y procede. El cliente ha determinado que el certificado del servidor es válido. Es responsabilidad del cliente ejecutar el siguiente paso o pasar directamente al paso final.

¿Coincide el nombre de dominio en el certificado del servidor con el nombre de dominio del propio servidor?

Este paso confirma que el servidor está localizado, en estos momentos, en la misma dirección de red que la que se especifica en el nombre de dominio del certificado del servidor.

Aunque este paso no es, técnicamente, parte del protocolo SSL, es la única forma de protección frente al ataque “man in the middle”.

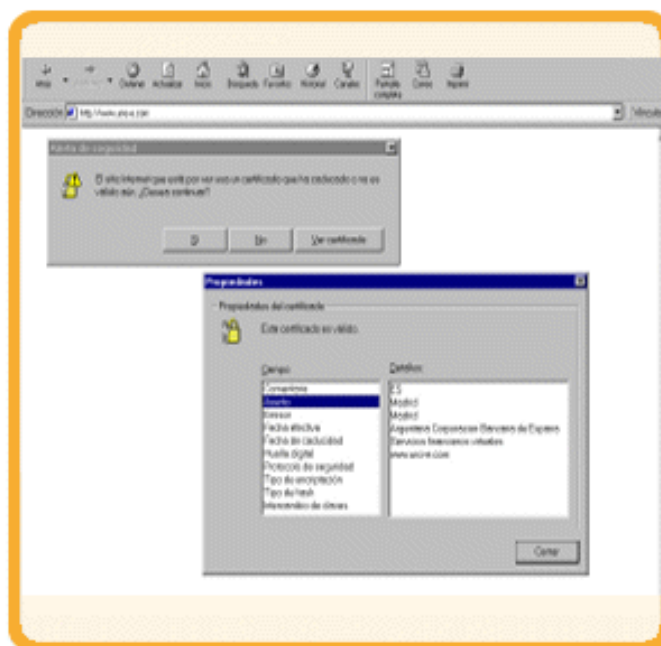
Los clientes que intenten este paso deben rechazar la autenticación del servidor o el establecimiento de conexión si no coinciden los nombres de dominio.

El servidor está autenticado.

El cliente procede ahora con el handshake SSL.

Si, por alguna razón, el cliente no llega aquí, el servidor identificado por el certificado no puede ser autenticado, y se avisará al usuario del problema y de que no se puede establecer una conexión encriptada y autenticada.

HTTPS (HTTP sobre SSL)



El protocolo SSL está, gracias a los esfuerzos de Netscape, ampliamente extendido.

La presencia de **https://** en el URL de un servidor indica se trata de un servidor "seguro" y que debe utilizarse SSL en la comunicación entre dicho servidor y cliente (navegador).

Los navegadores más extendidos (Netscape Navigator y Microsoft Internet Explorer) son capaces de "hablar" SSL.

¿Cómo se nos informa a los usuarios de que se va a entablar este procedimiento?

El navegador indicará que la zona es segura, mostrando un icono en la barra de estado (candado o llave) y, en caso de tener esta opción configurada en el navegador, aparecerá una ventana de aviso, o alerta de seguridad en donde le indicará que está a punto de ver las páginas bajo una conexión segura y que toda la información que intercambie con ese sitio no será vista por nadie en el web.

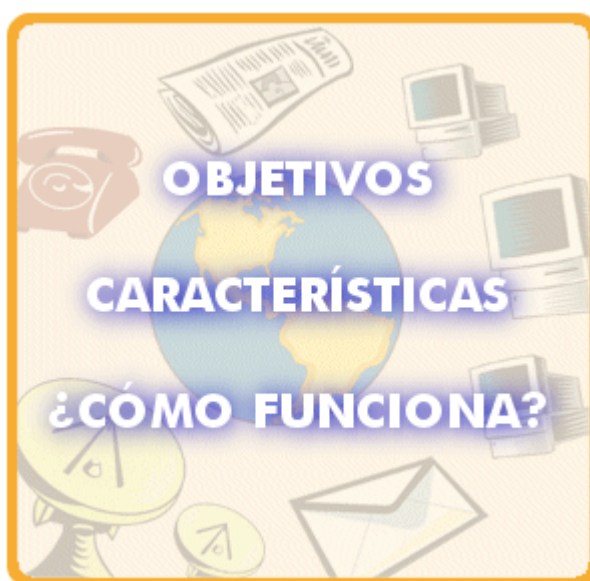
Igualmente informará de la existencia de un certificado y se podrá ver la información acerca del mismo.

6.3 El protocolo de seguridad para compras en Internet SET

Introducción a la Sección 6.3

Vas a comenzar el apartado 6.3:

Definición y características



¿Qué protocolo se utiliza en el famoso comercio electrónico? Pues el protocolo **SET (Transacción Electrónica Segura)**, que es un conjunto de normas o especificaciones de seguridad, encriptadas, que constituyen una forma/fórmula estándar para la realización de transacciones de pago a través de Internet.

Su objetivo está bastante claro: el protocolo SET autentifica los titulares de las tarjetas de crédito, los comerciantes y los bancos. Garantiza la confidencialidad de la información de pago y asegura que los mensajes no serán manipulados. Este protocolo ha sido desarrollado por Visa, Mastercard y otras empresas.

OBJETIVOS

Los objetivos del protocolo son:

- Proteger el sistema de tarjetas de crédito utilizado en Internet.
- Generar, en la mente del consumidor, una opinión de confianza respecto al nuevo concepto de Internet como mercado.
- Descubrir y aplicar nuevas transacciones financieras, seguras, para este nuevo canal.



CARACTERÍSTICAS

Sus principales características son: es un sistema abierto y multiplataforma, donde se especifican protocolos, formatos de mensaje, certificados, etc... sin limitación de lenguaje de programación, sistema operativo o máquina.

El formato de mensajes está basado en el estándar definido por la empresa RSA Data Security Inc. PKCS-7, como los protocolos S-MIME y SSL. El protocolo SET se puede transportar directamente en TCP, mediante correo electrónico con SMTP o MIME.

¿CÓMO FUNCIONA?

En SET se definen 5 agentes que pueden intervenir en transacciones comerciales:

- **Comprador.** Adquiere un producto utilizando la tarjeta de crédito.
- **Banco o entidad financiera.** Emite la tarjeta de crédito del comprador.
- **Comerciante.** Vende los productos.
- **Banco del comerciante.** Banco donde el comerciante tiene la cuenta.
- **Pasarela de pagos.** Gestiona la interacción con los bancos. Puede ser una entidad independiente o el mismo banco del comerciante.

Autenticación SET



¿Recordamos para qué sirve la autenticación?

En el proceso de transacción electrónica es un elemento muy importante, ya que la autenticación sirve para comprobar que los participantes en la operación comercial sean quienes dicen ser, es decir, que el consumidor sepa de modo totalmente seguro en qué comercio está comprando, y por otra parte, el comercio esté seguro de que quien está comprando sea realmente el titular del instrumento de pago.

¿Cómo se hace la autenticación en SET?

La autenticación se realiza a través de certificados digitales que tanto el comerciante como el comprador poseen, y que les son proporcionados por una tercera parte, la entidad financiera, como por ejemplo VISA.

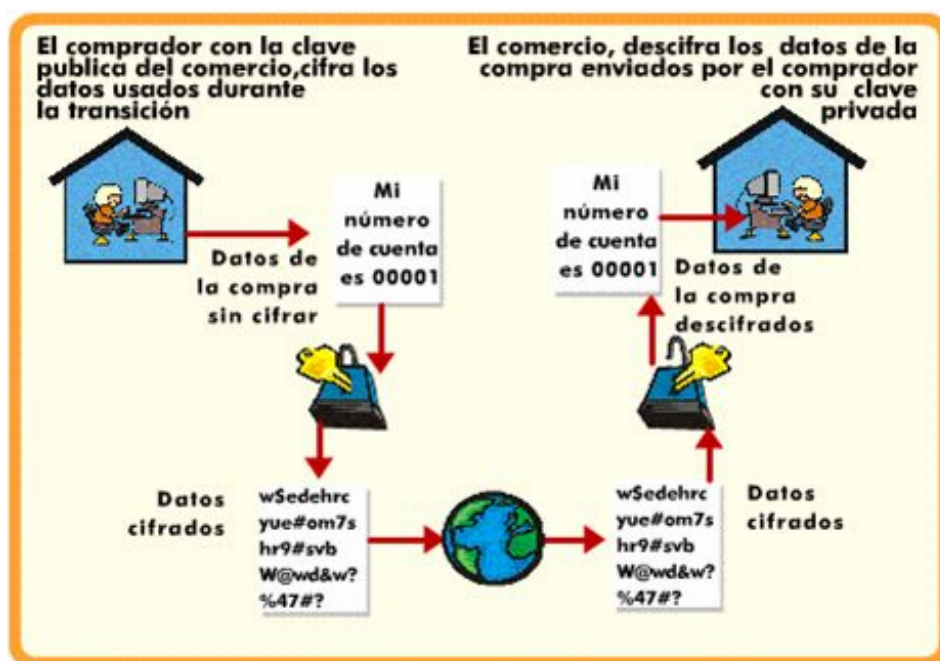
¿Y qué nos garantiza el certificado digital?

Principalmente, el certificado digital asegura la validez de una clave pública, e incluye los siguientes campos de información:

- ☒ Un identificador del propietario del certificado.
- ☒ Otro identificador de quién asegura su validez (que será una Autoridad Certificadora).
- ☒ Las fechas de inicio y caducidad del certificado.
- ☒ Un identificador del certificado (o número de serie).
- ☒ La clave pública del propietario del certificado.
- ☒ La firma digital de la Autoridad Certificadora, que asegura la autenticidad de todos los campos del certificado.

En definitiva, los certificados digitales sustituyen la función que realizan las tarjetas de crédito convencionales.

Privacidad SET



En toda comunicación, y en una transacción económica con más razón todavía, se desea que toda la información que viaja por la Red, durante el intercambio de identidades y datos, esté protegida contra cualquier intromisión o captura con métodos criptográficos, que cifran toda la información que se transmite entre las partes involucradas.

El procedimiento ya lo hemos visto en capítulos anteriores.

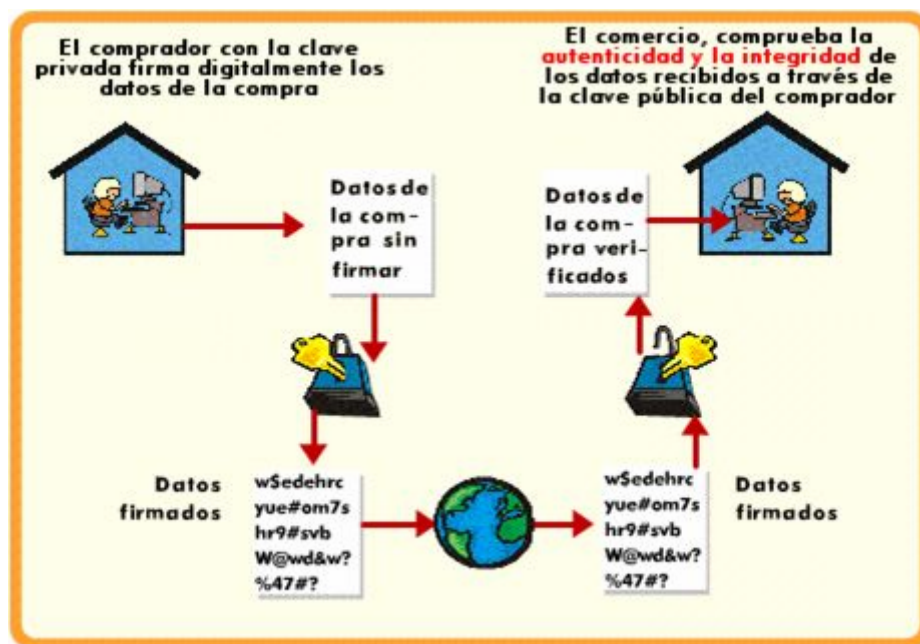
Los datos son encriptados a través de unos complejos algoritmos matemáticos prácticamente indescifrables.

En el caso del SET, hacemos uso de algoritmos que están dotados de dos claves asimétricas:

- Una clave pública, destinada a ser distribuida libremente para que los remitentes puedan cifrar sus datos.
- La otra clave, clave privada, sólo conocida por su legítimo propietario y custodiada con el máximo celo, sirve para descifrar los datos recibidos.

El mejor procedimiento es el que se puede observar en la ilustración, mediante el cual el comprador se asegura de que los datos de la transacción que acaba de realizar sean recibidos y leídos exclusivamente por el vendedor con el que la está realizando.

Integridad en SET



En SET, la integridad garantiza que los datos no han sido alterados de forma fraudulenta. La integridad, junto con la autenticidad, se basa en la generación de firmas digitales.

La firma digital se crea a partir de las relaciones matemáticas entre las claves pública y privada. Así, los datos cifrados con una de las claves sólo pueden ser descifrados con la otra. Pero en el caso de la firma digital, se invierten los papeles de las claves.

El procedimiento utilizado es el que ya hemos visto en capítulos anteriores: usando una función irreversible (MD5) se "destilan" los datos de la transacción, que luego son cifrados con la clave privada del remitente.

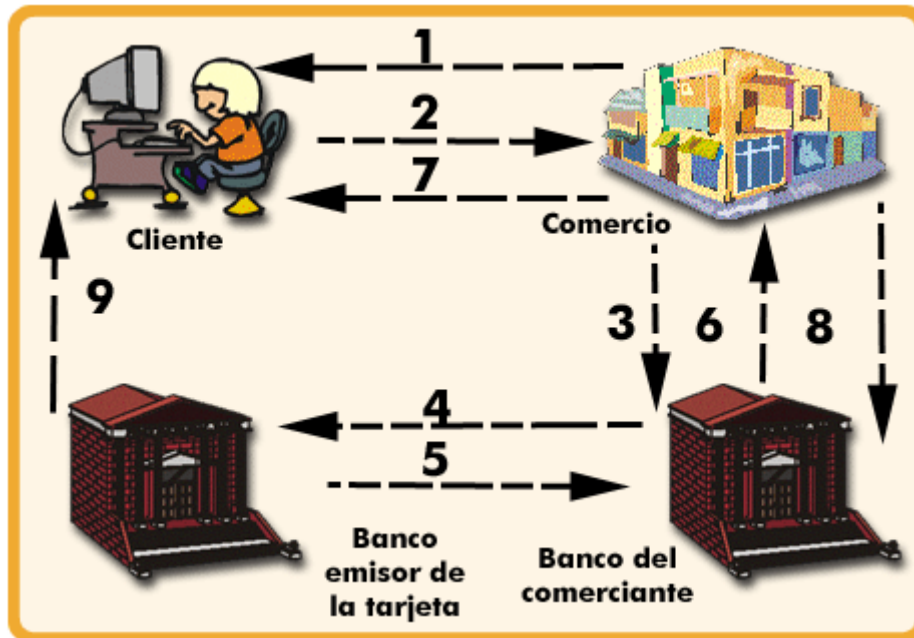
El resultado se añade al final del original que se envía, constituyendo así la firma digital del mismo.

El destinatario de los datos descifra el "destilado" a través de la clave pública del remitente.

Si el resultado del destilado es idéntico al original, la integridad y la autenticidad de los datos es correcta.

Si no son idénticos, significa que ha habido una manipulación no autorizada de los datos.

Funcionamiento de SET



El funcionamiento completo del protocolo SET es el que se muestra en la figura. Antes de verlo, es necesario hacer una aclaración: aparte de los elementos de la figura, existen dos agentes relacionados pero que no actúan directamente en las transacciones y son:

- **Propietario de la marca de la tarjeta.** Avalan las tarjetas: Visa, MasterCard, etc...
- **Autoridad de certificación.** Crea los certificados que se utilizan en las transacciones de la pasarela, el vendedor y el comprador. Pueden ser los bancos, los propietarios de la marca de la tarjeta o entidades independientes.

1. El cliente inicia la compra.
2. El cliente usando SET envía la orden y la información de pago al comerciante.
3. El comerciante pasa la información de pago al banco.
4. El banco verifica la validez del requerimiento.
5. El emisor de la tarjeta autoriza la transacción.
6. El banco del comerciante autoriza la transacción.
7. El servidor del comerciante complementa la transacción.
8. El comerciante captura la transacción.
9. El generador de la tarjeta envía el aviso de crédito al cliente.

6.4 Pretty Good Privacy (PGP)

Introducción a la Sección 6.4

Vas a comenzar el apartado 6.4:

Encriptación, compresión y firma digital usando PGP



PGP es un extendido software que proporciona los servicios de confidencialidad (mediante encriptación) y autenticación (usando firmas digitales). La versión más moderna es la 5.0 (PGPv5.0).

ENCRIPCIÓN

En cuanto a la encriptación, PGP emplea criptografía simétrica y asimétrica para proporcionar el servicio de confidencialidad. Como la mayoría de las aplicaciones actuales, el texto en claro es cifrado con una clave simétrica de sesión, la cual es posteriormente encriptada con una clave asimétrica. Las claves asimétricas empleadas tienen longitudes de hasta 4096 bits y se permiten los formatos RSA y DSS/Diffie-Hellman (por defecto).

Están disponibles tres cifradores simétricos de bloque: Triple-DES, CAST (su nombre viene de sus creadores: Carlisle Adams y Stafford Tavares de Northern Telecom - Nortel) e IDEA (International Data Encryption Algorithm), los cuales operan sobre textos en claro y textos cifrados de 64 bits. Triple-DES emplea claves de 168 bits, mientras que IDEA y CAST trabajan con un tamaño de clave de 128 bits.

Seguramente, Triple-DES te sonará de capítulos anteriores, aunque no ocurrirá así con CAST e IDEA, ¿no? CAST es un algoritmo nuevo, rápido y que parece inmune tanto a técnicas lineales como diferenciales de criptoanálisis, las cuales han conseguido romper DES. IDEA se sigue soportando por cuestiones de compatibilidad con versiones anteriores de PGP, pero es CAST el algoritmo empleado por defecto. Aunque CAST es más robusto que DES, el hecho de tener patente y no ser distribuido libremente ha hecho que no sea aceptado como un algoritmo estándar.



COMPRESIÓN

Por el lado de la compresión, PGP comprime el texto en claro antes de ser cifrado, lo que ahorra recursos a la hora de transmitir o almacenar el criptograma a la vez que mejora las características de encriptación, pues se consigue eliminar cualquier posible redundancia que pueda existir en dicho texto. PGP no comprime los textos cortos o los que no son susceptibles de una buena compresión y también reconoce los archivos que ya han sido tratados por los programas de compresión más populares, como PKZIP.

FIRMA DIGITAL

Para el tema de la firma digital, se emplea el algoritmo SHA-1 para generar la firma con las claves DSS. Por cuestiones de compatibilidad con versiones anteriores, se sigue soportando MD5 para realizar la firma con claves RSA.

6.5 IPsec

Introducción a la Sección 6.5

Vas a comenzar el apartado 6.5:

Definición y tipos



Comenzamos con algo de historia... Cuando los diseñadores de IPsec comenzaron sus trabajos, muchos ingenieros de Internet compartían el sentimiento de intentar lograr independencia en relación con el destino y la viabilidad de IPv4; así, muchos de ellos se vieron implicados en el desarrollo de IPv6.

Por lo tanto, IPsec es un híbrido: es, a la vez, producto de un “retrofit” a muchas de las experiencias adquiridas con IPv4, y una forma de protegernos frente a la inseguridad en IPv6.

Como muchos híbridos, tiene algunas peculiaridades inesperadas; pero a pesar de ellas, IPsec es el único estándar ampliamente aceptado para el transporte seguro a nivel de red en IP.

Es más versátil y más barato que cualquiera de las tecnologías de encriptación a nivel de aplicación y a nivel de enlace.

FORMADO POR...

IPsec está formado por un conjunto complejo de protocolos y mecanismos, y es importante entender sus componentes fundamentales y las formas en que se relacionan entre ellos.

Esta sección es una introducción a los componentes de IPsec, centrándonos en su integración en IPv4.



DISEÑADO PARA...

IPsec se diseñó para tener una arquitectura modular y abierta.

Esta modularidad le permite evolucionar frente a nuevos requerimientos, nuevas técnicas criptográficas y a los problemas identificados recientemente con los mecanismos de seguridad.

PECULIARIDADES

Una de las primeras peculiaridades de IPsec es que IPsec tiene dos protocolos de seguridad básicos: Cabecera de Autenticación (Authentication Header, **AH**) y Encapsulado de seguridad de la carga útil (Encapsulating Security Payload, **ESP**).

AH es un protocolo de autenticación que utiliza una firma digital hash en la cabecera del paquete para validar la integridad del paquete y la autenticidad del transmisor.

ESP es un protocolo de autenticación y encriptación que utiliza mecanismos criptográficos para proporcionar integridad, autenticación de la fuente y servicios de confidencialidad.

MODOS DE OPERACIÓN

Además, cada protocolo puede operar en uno de los dos modos siguientes:

- **Modo de transporte**, en el cual, el protocolo opera principalmente sobre la carga útil del datagrama original.
- **Modo túnel**, en el cual, el protocolo encapsula el datagrama original en otro nuevo, tratando el original como la carga útil del nuevo. Aquí, las direcciones IP fuente y destino son, a menudo pero no siempre, diferentes de las de el datagrama original.



Asociaciones de seguridad

¿Qué concepto se esconde aquí? Bueno, la definición nos dice que una asociación de seguridad es una relación en un solo sentido entre un emisor y un receptor.










Así pues, si se necesita una relación paritaria, para un intercambio seguro en dos sentidos, entonces se requieren dos asociaciones de seguridad. La asociación de seguridad le dice a un dispositivo IPsec cómo procesar los paquetes IPsec entrantes y cómo generar paquetes IPsec salientes.


En IPsec, una asociación de seguridad está identificada unívocamente por una dirección IP y un índice de parámetro de seguridad (SPI, Security Parameter Index).

Por tanto, en cualquier paquete IP, la asociación de seguridad está unívocamente identificada por la dirección destino de la cabecera y el SPI incluido en la cabecera de ampliación (cabecera de autenticación AH, o cabecera ESP).

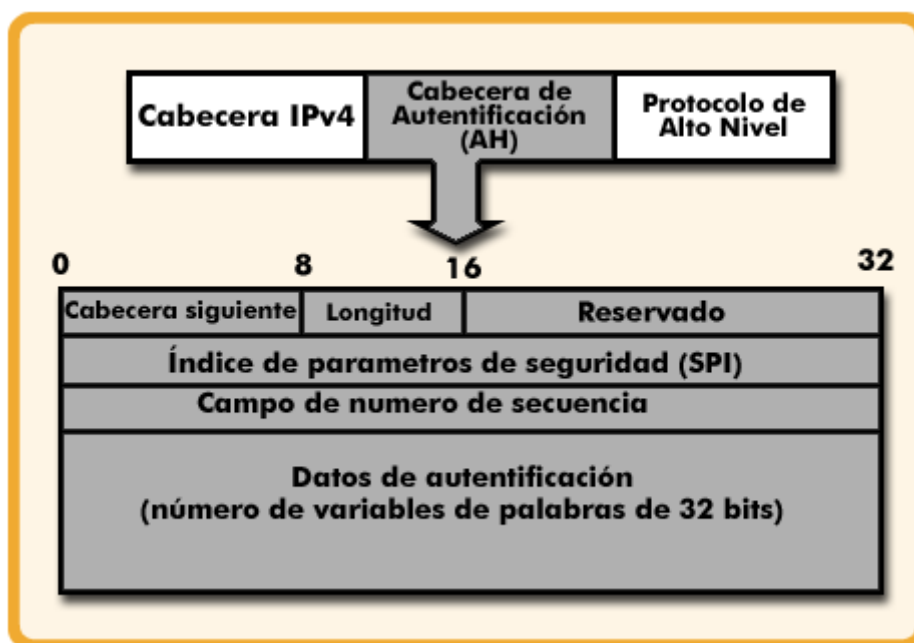
Asociaciones de seguridad

Una asociación de seguridad se define normalmente por los siguientes parámetros:

-  Algoritmo de autenticación y modo de utilización del algoritmo con la cabecera de autenticación de IP (requerido para implementaciones AH).
-  Clave(s) utilizadas con el algoritmo de autenticación en uso con la cabecera de autenticación (requerido para implementaciones AH).
-  Algoritmo de encriptado, modo del algoritmo y transformación que se está utilizando con el encapsulado IP de la carga de seguridad útil (requerido para implementaciones ESP).
-  Clave(s) usadas con el algoritmo de encriptado en uso con el encapsulado de seguridad de la carga útil (requerido para implementaciones ESP).
-  Presencia/ausencia y tamaño de la sincronización de criptografía o inicialización del campo vector para el algoritmo de encriptado (requerido para implementaciones ESP).
-  Clave(s) de autenticación usadas con el algoritmo de autenticación que es parte de la transformación ESP, si hay alguna (requerido para implementaciones ESP).
-  Tiempo de vida de la clave o tiempo en el que se debería cambiar la clave (recomendado para todas las implementaciones).
-  Tiempo de vida de la asociación de seguridad (recomendado para todas las implementaciones).
-  Dirección(es) origen de la asociación de seguridad; debería ser una dirección comodín, si existe más de un sistema que envía datos que comparten la misma asociación de seguridad con el destino (recomendado para todas las implementaciones).

 Nivel de seguridad (por ejemplo, secreto o no clasificado) de los datos protegidos (requerido para todos los sistemas que solicitan múltiples niveles de seguridad, recomendado para todos los otros sistemas).

Protocolo Authentication Header - AH (I)



Veamos ya el primero de los protocolos utilizados en IPsec.

El protocolo AH proporciona un medio para la integridad de los datos y la autenticación de los paquetes IP. Utiliza una cabecera de autenticación que consta de los campos de la imagen.

Cabecera siguiente (8 bits): identifica la cabecera que viene a continuación de ésta; generalmente TCP/UDP/ICMP para un datagrama IPv4.

Longitud (8bits): longitud del campo de datos de autenticación en palabras de 32 bits; será la longitud de la firma hash (nótese que este valor será constante para cada tipo de algoritmo hash, ya que una función hash siempre produce una salida de longitud fija).

Reservado (16 bits): para usos futuros.

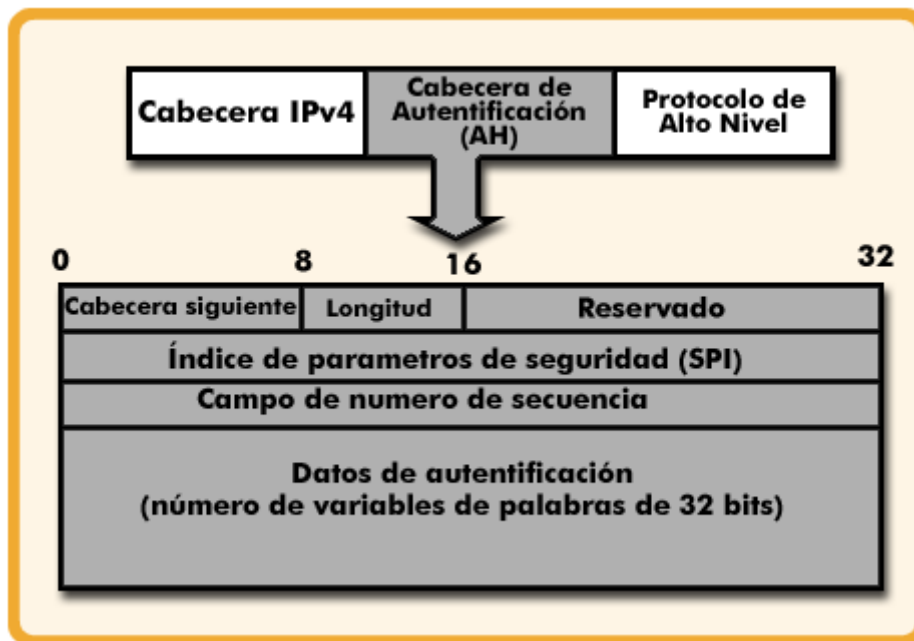
Índice de Parámetros de Seguridad (SPI, 32 bits): identifica una asociación de seguridad.

Número de secuencia: este campo, también llamado número de secuencia antireenvío, previene de las recepciones de paquetes ya recibidos, enviados generalmente por algún atacante.

Datos de autenticación (variable): se trata de la firma hash, un número entero de palabras de 32 bits.

Protocolo Authentication Header - AH (II)

Algunas notas complementarias son las siguientes:



El contenido del campo de datos de autenticación dependerá del algoritmo de autenticación especificado. En cualquier caso, los datos de autenticación se calculan utilizando el paquete IP entero.

Los cálculos de autenticación se llevan a cabo antes de la fragmentación en el origen y después de reensamblar en el destino. Por lo tanto, los campos relativos a la fragmentación se pueden incluir en los cálculos.

Normalmente, los pares IPsec (las partes comunicantes) utilizan MD5 o SHA-1 para crear una firma hash utilizando la clave de autenticación almacenada en la asociación de seguridad.

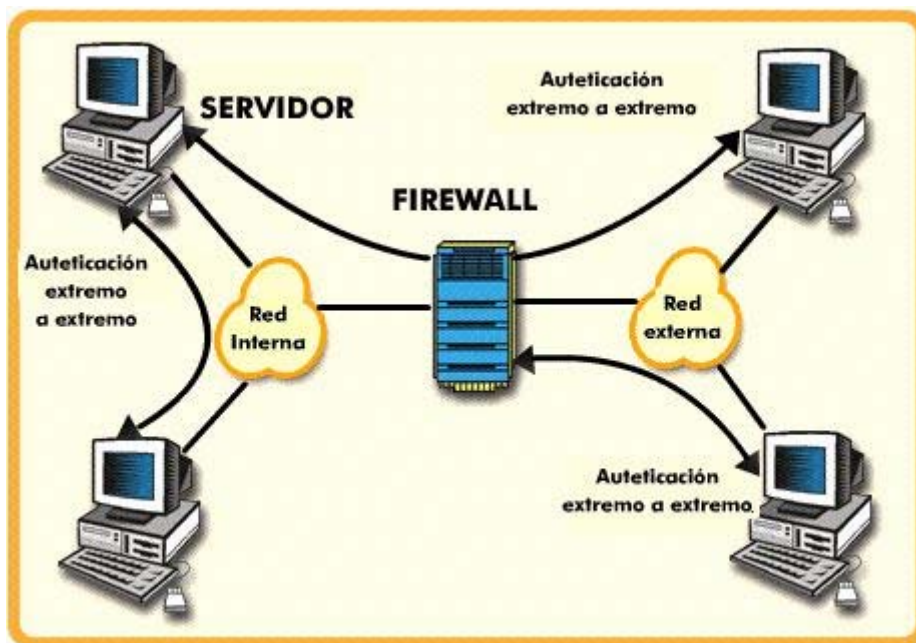
En IPv4, los campos tiempo de vida (TTL), tipo de servicio (ToS o DS) y suma de comprobación están sujetos a cambios, por tanto, se consideran cero en los cálculos de autenticación.

Las opciones de IPv4 se deben tratar de acuerdo a la regla de que si pueden cambiar durante su tránsito, no se deben incluir en los cálculos.

En IPv6, el campo límite de saltos es el único campo en la cabecera base IPv6 sujeta a cambios; por tanto, se considera cero en los cálculos.

Para las cabeceras opciones salto-a-salto y opciones del destino, el campo tipo de opción de cada opción contiene un bit que indica si el campo de datos de opción puede cambiar durante el tránsito; si es así, esta opción se excluye de los cálculos de autenticación.

Autenticación utilizando claves MD5



¿Por qué utilizamos MD5 para la autenticación? Pues porque la RFC 1828 especifica el uso de la función hash MD5 para autenticación.

El algoritmo MD5 se implementa en el origen sobre el paquete IP, conjuntamente con una clave secreta, y después se inserta en el paquete IP.

En el destino, se llevan a cabo los mismos cálculos sobre el paquete IP conjuntamente con una clave secreta y se compara con el valor recibido.

Este procedimiento proporciona autenticación e integridad de los datos.

Específicamente, los cálculos MD5 se llevan a cabo con la secuencia siguiente:

clave, relleno de clave, paquete IP, clave, relleno de MD5.

donde cada uno de estos campos significa:

Clave = la clave secreta para la asociación de seguridad.

Relleno de clave = relleno para que la clave + relleno sea múltiplo entero de 512 bits.

Paquete IP = paquete con los campos apropiados considerados cero.

Relleno MD5 = relleno proporcionado por MD5 para que el tamaño del bloque completo sea múltiplo entero de 512 bits.

Además, existen dos formas en las que se puede realizar el servicio de autenticación de IP.

En un caso (extremo - a - extremo), la autenticación se realiza directamente entre un servidor y una estación de trabajo cliente.

La estación puede estar en la misma red que el servidor o en una red externa.

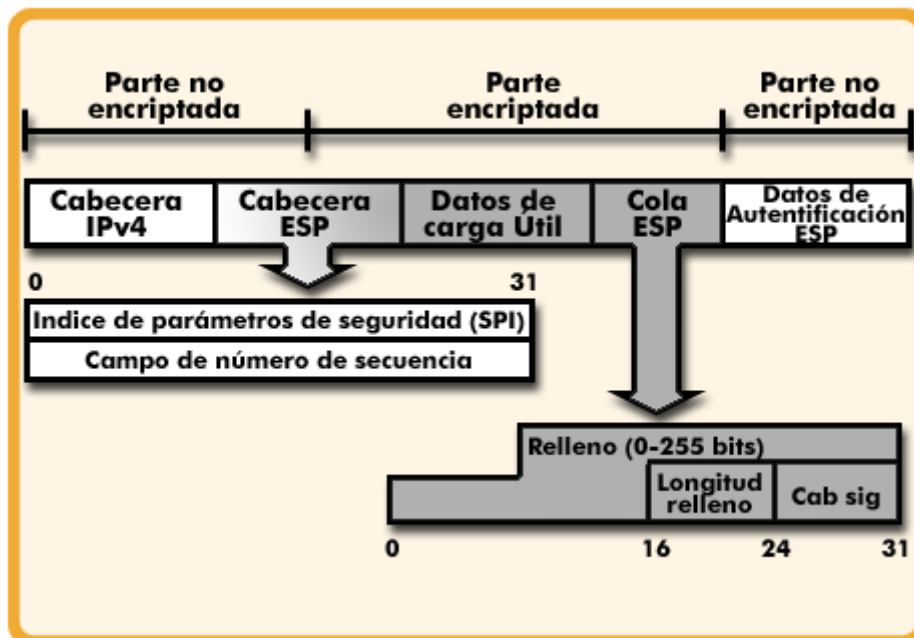


Mientras la estación y el servidor compartan una clave secreta protegida, el proceso de autenticación es seguro.

En el otro caso (extremo - a - intermedio), una estación de trabajo remota se autentica por medio de un firewall corporativo, creado para controlar el acceso a la red interna completa o porque el servidor requerido no permite la autenticación.

Encapsulado de seguridad de carga útil (ESP)

¿Y si continuamos ahora con el segundo protocolo utilizado en IPsec?



El uso del encapsulado de seguridad de carga útil proporciona privacidad e integridad de los datos de los paquetes IP, que es lo que estamos buscando.

Dependiendo de los requisitos del usuario, este mecanismo se puede utilizar para encriptar bien el segmento de la capa de transporte (por ejemplo, TCP, UDP, ICMP), conocido como *modo de transporte ESP* o bien el paquete IP completo, conocido como *modo túnel ESP*.

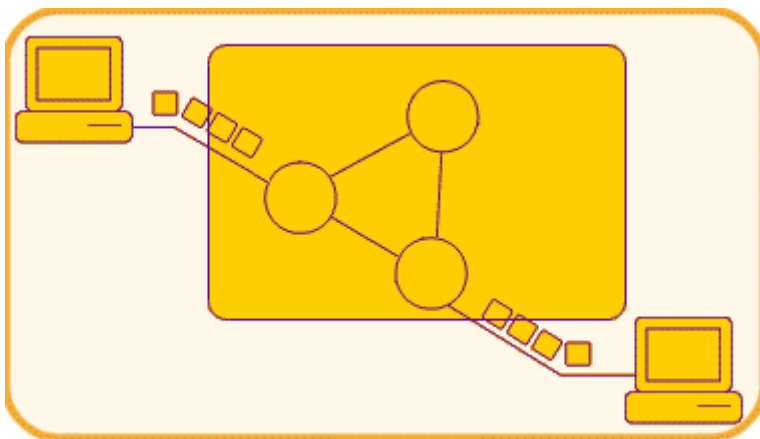
Y en concreto, el protocolo lo que hace es añadir una información adicional al datagrama IP mediante la cabecera correspondiente.

La cabecera ESP comienza con un índice de parámetros de seguridad (SPI) de 32 bits, que identifica una asociación de seguridad y un número de secuencia, también de 32 bits, para la prevención de los ataques de reenvío.

El resto de la cabecera, si existe, puede contener parámetros dependiendo del algoritmo de encriptado que se está utilizando.

En general, la primera parte de la cabecera, incluyendo el SPI, el número de secuencia y, posiblemente, algunos parámetros, se transmiten sin encriptado (texto en claro), mientras que el resto de la cabecera, si existe, se transmite encriptado.

Envío y recepción de un paquete ESP



En la transmisión y recepción de paquetes ESP se opera de la siguiente forma.

TRANSMISIÓN

Para transmitir un paquete ESP el host o el gateway que lo envía debe operar como sigue:

- ☒ Identificar la asociación de seguridad (SA) correspondiente y el SPI asociado. Si el SA indica integridad/autenticidad, el origen debe localizar el algoritmo hash y la clave secreta adecuadas. Si la SA indica confidencialidad, el origen debe localizar la transformación criptográfica y la clave apropiadas.
- ☒ Incrementa el contador anti-repetición (si la SA indica que hay que utilizarlo).
- ☒ Ensambla la carga útil ESP y el relleno, si éste existe.
- ☒ Si se especifica en la SA, encripta el resultado del ensamblado anterior con el algoritmo criptográfico apropiado.
- ☒ Si la SA lo especifica, calcula el valor de chequeo de integridad (ICV) sobre las cabeceras y carga útil (menos el propio campo de autenticación), utilizando el algoritmo hash apropiado.
- ☒ Inserta la cabecera ESP, carga útil y cola (más el campo de autenticación/integridad, si éste se requiere) directamente después de la cabecera IP.

RECEPCIÓN

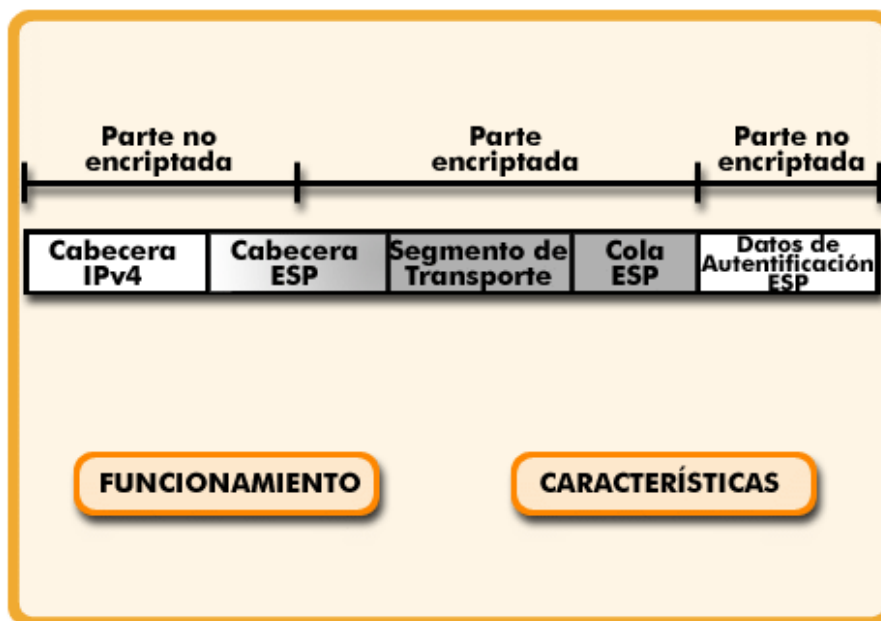
Y en el receptor, host o gateway, se invierte el proceso de envío descrito en la sección anterior. Si la SA especifica servicios de autenticación, se requiere un chequeo hash similar al que se efectúa con AH: el receptor debe descartar cualquier paquete que falle en el test de integridad. Si la SA especifica opción anti-respuesta, el receptor debe descartar cualquier paquete que está repetido o cuyo número de



secuencia se salga de los valores de la ventana de recepción. El último paso es desenscriptar el paquete.

Si el ICV no está presente, IPsec no podrá detectar la modificación de la carga útil, pero el producto de la desenscripción será, probablemente, inentendible. Un ICV proporciona un mecanismo de descarte de datos modificados, con una sobrecarga mínima.

Modo transporte ESP



¿Cómo enviamos los datagramas IP con este protocolo? Mediante el modo transporte ESP, que se utiliza para encriptar datos transportados por IP. Normalmente, estos datos son segmentos de la capa de transporte, como segmentos TCP o UDP, que, a su vez, contienen datos de la capa de aplicación. Para este modo, la cabecera ESP se inserta en el paquete IP justo antes de la cabecera de la capa de transporte (por ejemplo, TCP, UDP, ICMP). En el caso de IPv6, si la cabecera de opciones del destino está presente, la cabecera ESP se inserta justo antes de esa cabecera.

El **funcionamiento** en modo transporte puede resumirse como sigue:

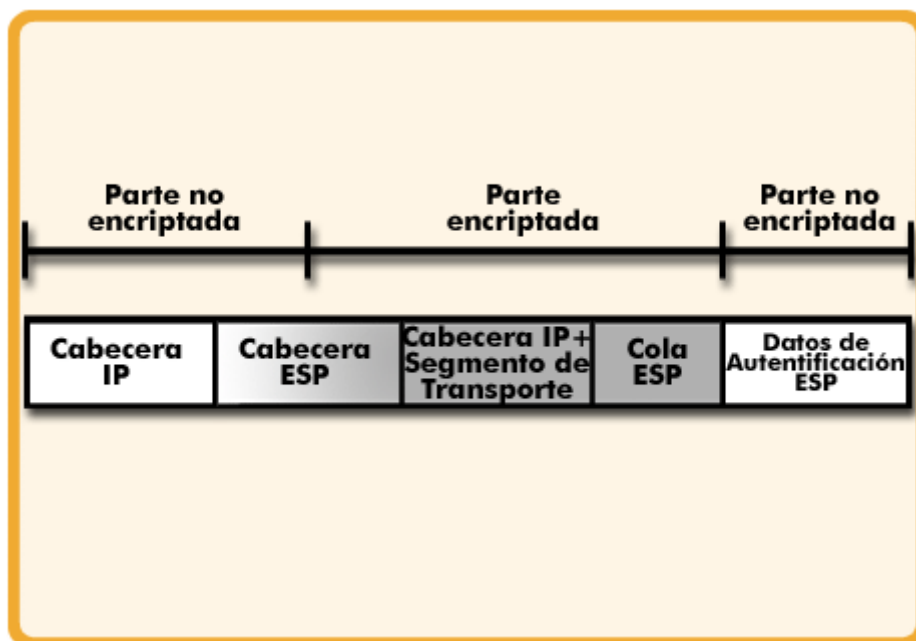
- En el origen, el bloque de datos que consta de la cola ESP mas el segmento entero de la capa de transporte, se encripta y el texto en claro se reemplaza con el texto cifrado para formar el paquete IP que se va a transmitir.
- Este paquete es, entonces, encaminado al destino. Los router intermedios no necesitan examinar el texto cifrado.
- El nodo destino examina y procesa la cabecera IP mas cualquier cabecera de ampliación IP en texto en claro (IPv6). Entonces, sobre la base del SPI, en la cabecera ESP, el nodo destino chequea la integridad del paquete (si se requiere) y, finalmente, desencripta el resto del paquete para recuperar el segmento de transporte en texto en claro.

Resumiendo, ¿cuáles son sus **características**?

La operación en modo transporte proporciona privacidad para cualquier aplicación que la utilice, evitando así la necesidad de implementar privacidad en cada aplicación individual.

Este modo de operación es también razonablemente eficiente, añadiendo poco a la longitud total del paquete IP. Un inconveniente de este modo es que es posible hacer un análisis de tráfico con los paquetes transmitidos.

Modo túnel ESP



Entonces, ¿hay alguna forma de solucionar los inconvenientes del modo transporte ESP?

Pues sí, utilizando el modo túnel ESP.

El modo túnel ESP se utiliza para encriptar el paquete IP entero. Para este modo, la cabecera ESP se incorpora como prefijo al paquete y, después, el paquete mas la cola ESP, y se encripta todo desde la parte encriptable de la cabecera ESP.

Este método se puede utilizar para impedir el análisis de tráfico.

Ya que la cabecera IP contiene la dirección de destino, y posiblemente directivas de encaminamiento del origen e información de opciones salto - a - salto, no es posible simplemente transmitir el paquete IP encriptado con prefijo la cabecera ESP.

Los routers intermedios no serían capaces de procesar ese paquete.

Por tanto, es necesario encapsular el bloque completo (cabecera ESP mas el paquete IP encriptado) con una nueva cabecera IP, que contendrá suficiente información para el encaminamiento, pero no para el análisis de tráfico.



Comparación entre métodos

Para acabar con el tema, si comparamos los dos métodos ESP existentes, podemos concluir lo siguiente:

Mientras el modo de transporte es conveniente para proteger las conexiones entre ordenadores que incorporan la característica ESP, el modo túnel es útil en una configuración que incluya firewalls u otro tipo de pasarela de seguridad que proteja una red confidencial de redes externas.

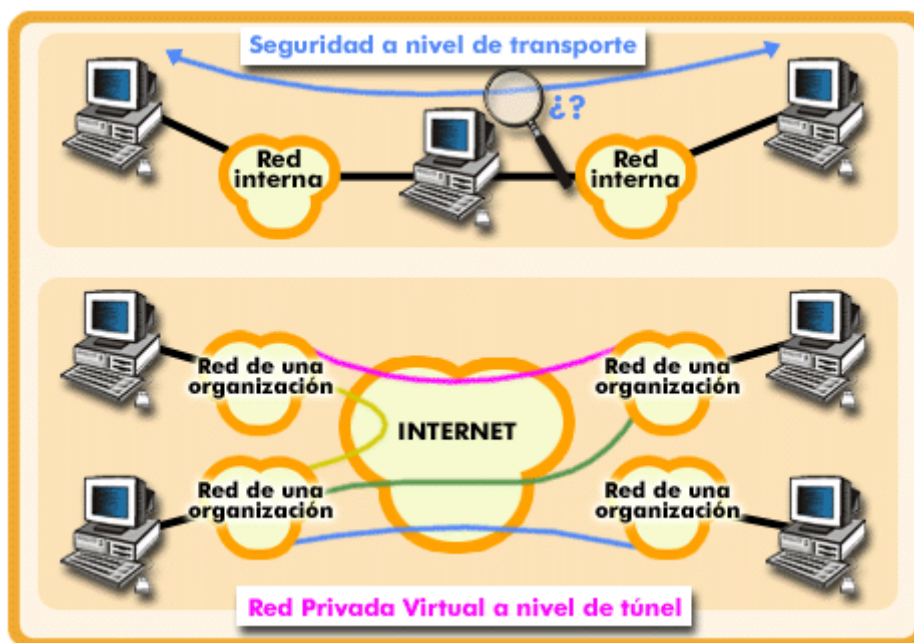
En este último caso, el encriptado ocurre solamente entre un ordenador externo y la pasarela (gateway) de seguridad, o entre dos pasarelas de seguridad.

Esto libera a los ordenadores de una red interna de la carga de procesamiento del encriptado, a la vez que simplifica la tarea de distribución de claves, debido a la reducción del número de claves necesarias.

Además, impide el análisis de tráfico basado en el destino último.

Aplicaciones ESP

¿Te apetece clarificar un poco los conceptos vistos con dos ejemplos? Pues vamos a ello:



Considera el caso en el que un ordenador externo desea comunicarse con un ordenador en una red interna protegida por un firewall, y en el que el ESP se implementa entre ordenadores externos y el firewall.

Durante la transferencia de un segmento de transporte entre un ordenador externo y otro interno, tienen lugar los siguientes pasos:

- El origen prepara un paquete IP interno con la dirección de destino del ordenador destino interno.

Al paquete se le incorpora una cabecera prefijo ESP, después, el paquete y una parte de la cabecera ESP se encriptan. El bloque resultante se encapsula con una nueva cabecera IP (cabecera base mas ampliaciones opcionales tales como encaminamiento y opciones salto - a - salto) cuya dirección destino es la del firewall; esto forma el paquete exterior.

- El paquete exterior se encamina al firewall de destino.

Los router intermedios necesitan examinar y procesar la cabecera IP externa mas las cabeceras de ampliación IP externas, pero no necesitan examinar el texto cifrado.

- El firewall destino examina y procesa la cabecera IP externa mas cualquier cabecera de ampliación IP externa.

Entonces, sobre la base del SPI y la cabecera ESP, descripta el resto del paquete para recuperar el texto en claro del paquete IP interno.

Este paquete se transmite después a la red interna.

- El paquete interno se encamina a través de cero o más routers de la red interna hasta alcanzar al ordenador de destino.

También, la operación en modo túnel se puede utilizar para establecer una red privada virtual.

En este ejemplo, una organización tiene cuatro redes privadas interconectadas a través de Internet.

Los ordenadores en la red interna utilizan Internet para transportar datos, pero no interactúan con otros ordenadores de Internet.

Los túneles terminan en las pasarelas de seguridad de cada red interna, lo que permite a los ordenadores evitar la implementación de las capacidades de seguridad.

Utilización de AH frente a ESP



¿Por qué preocuparnos por AH?

Aunque en un principio no fue así, el estándar final de ESP soporta tanto la autenticación y la integridad como la privacidad; por tanto, podríamos preguntarnos para qué necesitamos implementar AH, que sólo proporciona los mecanismos de autenticación e integridad.

Sin embargo, existen razones prácticas e históricas para seguir utilizando AH en algunos servicios:



Cualquier implementación que se declare conforme al estándar ESP debe soportar, obligatoriamente, el algoritmo de encriptación DES, incluso aunque su única pretensión sea utilizar los mecanismos de autenticación e integridad de DES y no vaya a utilizar jamás los mecanismos de encriptación.

Las leyes reguladoras de algunos países solo permiten la utilización de los mecanismos de encriptación para propósitos de autenticación, no de confidencialidad, por lo tanto, no merece la pena incorporar DES si no lo vamos a usar en toda su potencia.

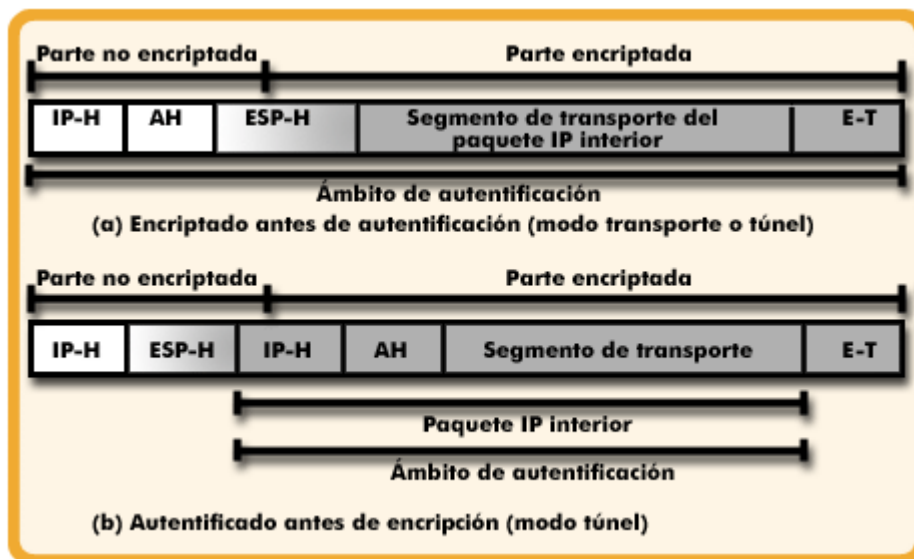


AH proporciona un nivel más alto de seguridad de la autenticidad e integridad de los contenidos que ESP, ya que opera con la carga útil del datagrama IP más todos los campos invariables de la cabecera IP; mientras que ESP sólo opera con la carga útil del datagrama IP.

Utilización conjunta de AH y ESP

Y ahora una potencialidad adicional de IPsec: los dos mecanismos de seguridad de IP se pueden combinar para transmitir un paquete IP que tenga autenticación AH más privacidad ESP (y si queremos, autenticación ESP).

Existen dos técnicas que se pueden utilizar, diferenciadas por el orden en el que se apliquen los dos servicios (autenticación y privacidad). Por simplicidad, suponemos que no aplicamos también autenticación ESP.



ENCRYPTADO ANTES DE AUTENTICACIÓN

La primera de las dos técnicas posibles es el caso del **encryptado antes de la autenticación**. En este caso, el paquete IP entero transmitido se autentica, incluyendo ambas partes, la encryptada y la no encryptada. En esta técnica, el usuario primero aplica ESP a los datos que se van a proteger, después incorpora al principio la cabecera de autenticación y la cabecera IP en texto claro. Realmente, existen dos subcasos:

- **ESP en modo transporte:** la autenticación se aplica al paquete IP entero entregado al destino, pero solamente el segmento de transporte se protege por el mecanismo de privacidad (encryptado).
- **ESP en modo túnel:** la autenticación se aplica al paquete IP entero entregado a la dirección IP destino externa (un firewall, por ejemplo), y la autenticación se lleva a cabo en el destino. El paquete IP interno se protege por el mecanismo de privacidad, para su entrega al destino IP interno.

AUTENTICADO ANTES DE ENCRIPCIÓN

La segunda de las técnicas posibles constituye el caso en el que la **autenticación se aplica antes del encryptado**. Esta técnica sólo es apropiada para ESP en modo túnel. En este caso, la cabecera de autenticación se sitúa dentro del paquete IP interno. Este paquete interno es autenticado y protegido por el mecanismo de privacidad.

Como hemos visto, las funciones de autenticación y encriptado se pueden aplicar en cualquier orden para ESP en modo túnel. El uso de la autenticación antes del encriptado puede ser preferible por varias razones. Primero, ya que AH se protege por ESP, es imposible que cualquiera intercepte el mensaje y altere AH sin ser detectado. Segundo, puede ser deseable almacenar la información de autenticación con el mensaje y el destino para una referencia posterior. Es más conveniente hacer esto si la información de autenticación se aplica a un mensaje no encriptado; de otra forma, el mensaje tendría que ser reencriptado para verificar la información de autenticación.



Bienvenido al capítulo:

Dispositivos de seguridad



7.1 Firewalls

Introducción a la Sección 7.1

Vas a comenzar el [apartado 7.1](#):

¿Qué es un firewall?

Los dispositivos más ampliamente utilizados para proporcionar seguridad a las redes de datos son los **cortafuegos** o **firewalls**.

¿Pero, qué es un firewall?. Básicamente, un firewall es un dispositivo que se coloca entre las redes internas de un usuario (corporativo o residencial) y el mundo exterior.

Los firewalls han venido siendo utilizados durante mucho tiempo en grandes redes públicas, y son un punto de partida para el desarrollo de una estrategia de seguridad.

La razón para utilizarlos como punto de partida de la estrategia de seguridad es que, generalmente, se colocan en los puntos en los que la red privada se conecta a una red pública, como por ejemplo, Internet.

Aunque por sí sólo no constituye una estrategia de seguridad perfecta, es muy fácil de configurar; tan sólo requiere la modificación de un router de pasarela (de un **gateway**).

Por supuesto, si disponemos de varios puntos de conexión a Internet, necesitaremos un firewall por cada uno de dichos puntos. La complejidad de este proceso se ve dramáticamente incrementada cuando el número de puntos de conexión aumenta.

Funcionalidades de los firewalls



Hagamos un poco de historia, ¿no te parece? El Departamento de Defensa de Estados Unidos, probablemente la mayor autoridad mundial en sensibilidad de datos y controles de seguridad, utilizaba un sistema confidencial definido como niveles de seguridad para el acceso restringido a documentos clasificados.

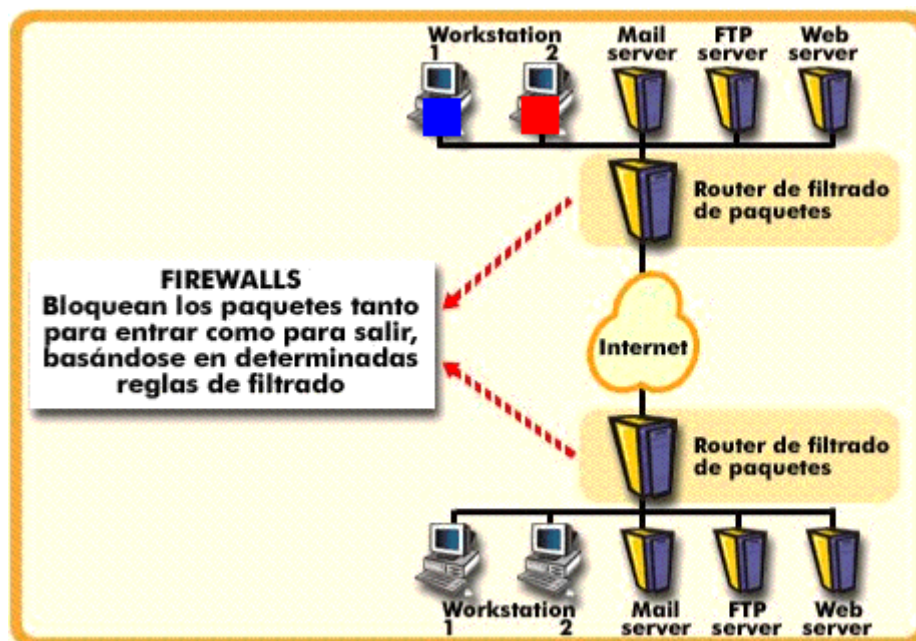
El criterio para determinar cómo debería protegerse una entidad gubernamental estaba detallado en el legendario "Libro Naranja", donde se establecía que los datos susceptibles de clasificarse como de alta seguridad nunca deben residir en ordenadores que posean conexiones al exterior de la red.

Por supuesto, ésta es la mejor estrategia "cortafuegos" que existe, pero es demasiado restrictiva para ser práctica. Todo el mundo es consciente, hoy día, de las ventajas de la interconectividad y de que si queremos aislar datos con niveles altos de seguridad, lo mejor que podemos hacer es aislar al ordenador que los posea de cualquier tipo de conectividad.

Así, para un administrador de red, los firewalls cumplen dos funciones básicas.

- Controlar las máquinas que un elemento externo a la red puede ver, y los servicios de dichas máquinas con las que dicho elemento externo puede dialogar.
- Controlar las máquinas de Internet que un elemento interno a la red privada puede ver, así como los servicios que puede utilizar de éstas.

Características de los firewalls



Por lo tanto, un firewall es mucho más que un “policía de tráfico” que organiza la ruta que puede seguir un tipo de tráfico y decide a qué tráfico debe detener.

Y es que, en general, un firewall tiene las siguientes funcionalidades:

- Es un medio que sirve para regular el acceso a la red de ordenadores de una organización.
- Controla el acceso y registra los intentos de acceder a la red.
- Decide permitir o no la comunicación, de acuerdo con la política de seguridad configurada.
- Un firewall para Internet se suele instalar en el punto donde la red interna sale a Internet.
- Frecuentemente, está constituido por un conjunto de elementos hardware.

La utilización de firewalls es algo muy conveniente para la seguridad de nuestra red.

Sin embargo, por el mero hecho de disponer de ellos, no tenemos garantizada la seguridad absoluta; hay que tener cuidado con las **“puertas de atrás”** que pueden abrirse en nuestras redes.

De nada nos valdrá haber adquirido el mejor firewall y tenerlo colocado en el punto estratégico de entrada y salida de la red, si permitimos que los usuarios de nuestra red puedan utilizar, por ejemplo, accesos privados por marcación para llegar a Internet.

En ese caso, tendríamos una puerta abierta al mundo exterior, totalmente incontrolada por el gateway / firewall.

7.2 Tipos de firewalls

Introducción a la Sección 7.2

Vas a comenzar el apartado 7.2:

Routers que restringen paquetes o filtros de paquetes



Cuando hablamos de un firewall, podemos distinguir entre varios tipos de los mismos. Empecemos por los más “intuitivos”, si no te importa...

Los firewalls de filtrado de paquetes constituyen la primera generación de firewalls. Controlan el filtrado de paquetes deciden el envío de tráfico hacia una red basándose en una serie de reglas predefinidas.

Los filtros de paquetes presentan dos importantes ventajas: son fáciles de implementar (generalmente no requieren dispositivos hardware adicionales a los router ya instalados) y son transparentes a los usuarios (cosa que no sucede con todos los métodos de firewall).



Un router no toma ninguna decisión basándose en los contenidos de la “carga útil” (payload) del paquete, sino basándose en quién lo envía y hacia quién va destinado.

Sólo considera que si el paquete concuerda con un determinado número de parámetros (direcciones IP y puertos), debe tomar una acción apropiada para que el paquete pase a su través o se le deniegue dicho paso.

Estas tablas que conceden el paso o lo deniegan se configuran de acuerdo a las políticas globales de seguridad de red establecidas por el administrador de red o el coordinador de seguridad.



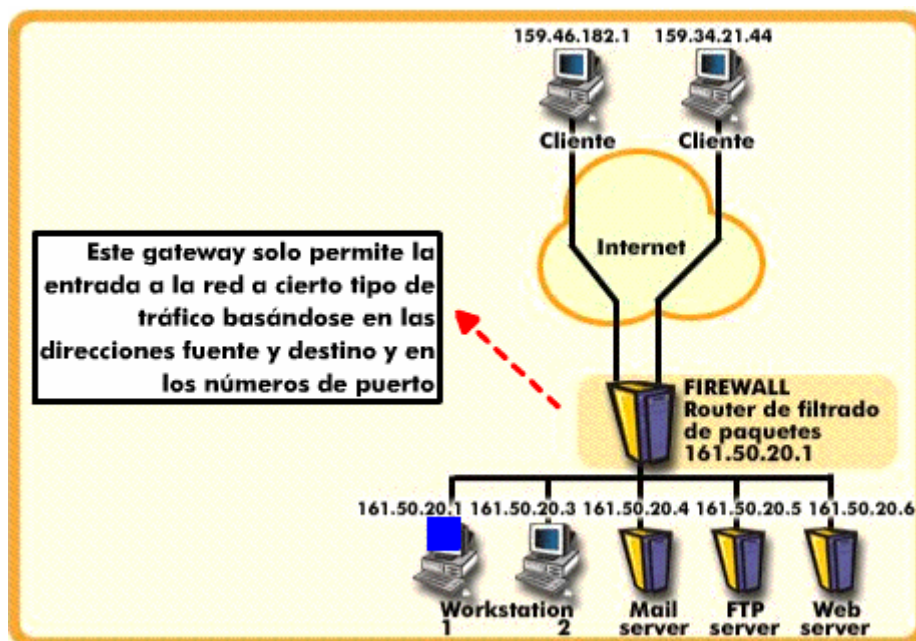
Básicamente, un filtro de paquetes puede tomar dos formas diferentes. La primera es una red abierta con un filtrado selectivo del tráfico no deseado.

Para cada tipo de ataque a la red, se debe utilizar un filtro específico en el router. La segunda es una red cerrada con filtrado selectivo del tráfico deseado.



Aunque ofrecen una mayor seguridad, incluso frente a ataques en los que ni siquiera hemos pensado, se requiere un gran esfuerzo de los administradores de red para actualizar el firewall cuando se añaden o modifican nuevos ordenadores o nuevos servicios.

Características del filtro de paquetes



Enseguida nos damos cuenta de que un filtro de paquetes presenta algunas carencias. La primera de todas es que no existe ninguna forma de autenticar a los usuarios.

Por ejemplo, no hay forma de que podamos configurar quién es el que puede o no puede mandar el correo. Así, no hay manera de configurar en el firewall que un determinado empleado de una compañía pueda consultar su correo en el servidor, independientemente de la dirección IP desde la que esté trabajando.

Después, por razones de rendimiento, los routers actuales no “abren” todos los paquetes que reciben. En el futuro, aunque ya hoy en día hay soluciones en el mercado, lo que se plantea es un cambio en el comportamiento de los gateways; habrá una división entre equipos de encaminamiento y de filtrado de paquetes.

Ya están surgiendo nuevos productos que soportan mecanismos de autenticación dinámica a nivel de usuario en los routers de filtrado de paquetes, incluso con comunicaciones encriptadas.

Por último, es muy habitual que los cambios en la configuración de las redes requieran reconfiguraciones a gran escala de los router-gateways y de los firewall de filtrado de paquetes.

Esto puede llevar bastante tiempo y llegar a ser un generador potencial de desastres si no se es muy cuidadoso a la hora de reconfigurar; pensemos que “pequeños” descuidos pueden dejar abiertos grandes tramos de red o dejar a los routers incapacitados para enviar paquetes, incumpliendo así su primera tarea como directores del tráfico.

Host bastión

Otro tipo de firewall es el host bastión.

Un host bastión o un host de filtrado, como también se les denomina, utiliza los mecanismos de filtrado que ofrecen los routers mas un host "asegurado".

Un host asegurado es aquel que ha "sufrido" un "peinado", por parte de un experto en seguridad, en su sistema operativo y sus servicios principales, y que se encuentra en la red interna.

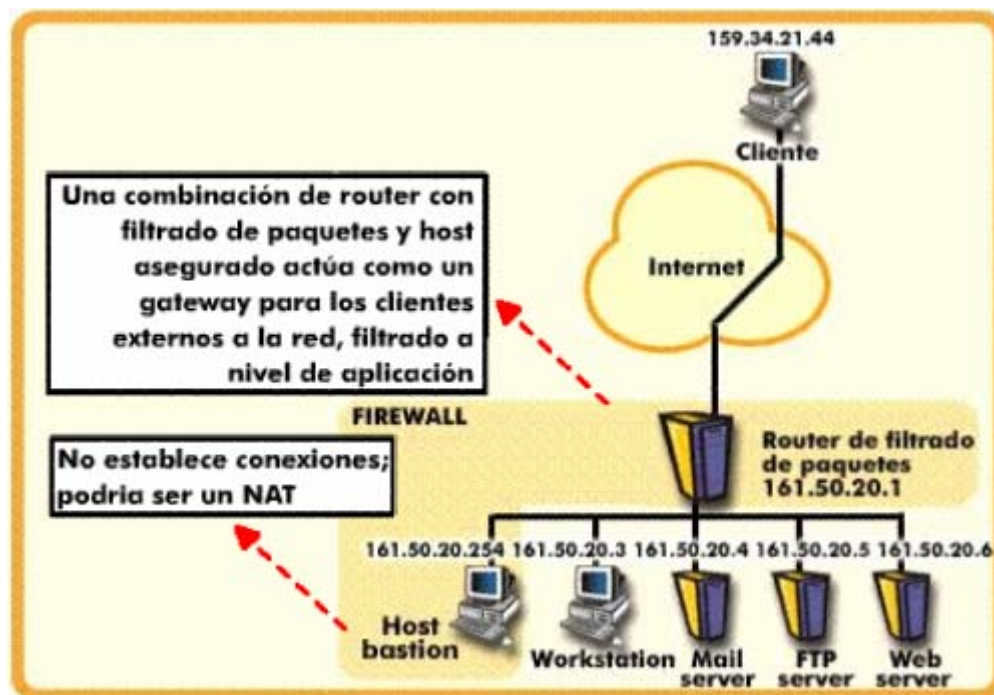
La primera barrera de seguridad la proporciona el filtro de paquetes; el host asegurado se utiliza para reconducir los flujos de información en la dirección adecuada.

El host bastión es una máquina segura que está conectada a Internet de la misma forma que las demás, sólo que el gateway permite que el tráfico se dirija hacia él de una forma menos restrictiva.

Los host bastión suelen utilizarse en combinación con los router de filtrado debido a que los sistemas simples de filtrado no pueden filtrar a nivel de protocolo o de aplicación.

El host bastión es el principal punto de contacto para las conexiones que entran desde el mundo exterior.

Características del host bastión



Un host bastión es mucho más fácil de configurar y de mantener que un servidor distribuido, ya que el grueso del tráfico sólo es enviado a un sistema.

Puesto que el host bastión está situado en el interior del cableado, no requiere de consideraciones especiales con respecto a otros equipos conectados localmente.

La política de seguridad de cada sitio determinará lo que es necesario configurar en el filtro de paquetes, que puede ser tan restrictivo como sea necesario.

Una de las grandes ventajas de la utilización de este esquema de seguridad es que el filtro de paquetes puede tener una configuración extremadamente simple; tan simple como una sentencia genérica de "denegar todo", precedida de unas pocas sentencias de "permiso" que pertenecen sólo al host bastión.

En grandes y cambiantes redes, este procedimiento reduce la carga del personal de seguridad; añadir nuevas máquinas o disponer de una pobre equipación de seguridad en los usuarios no afecta al firewall ni a la protección que ofrece el host bastión.

Por supuesto, el hecho de centralizar todo el control en un punto tiene también sus desventajas.

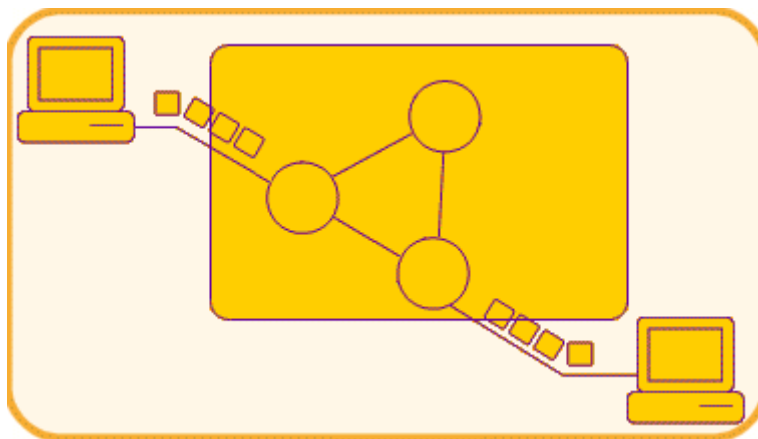
Redes grandes y sobrecargadas necesitarán varias máquinas que actúen como host bastión (lo cual vuelve más compleja su administración), o incluso una red perimetral de hosts bastión, como veremos a continuación.

Cada máquina necesita su propia sección en el firewall de filtrado de paquetes, incrementando la complejidad.



Otra gran desventaja es que, por el hecho de centralizar todo el control en un punto, si alguien se hace con los privilegios de administración de dicha máquina, podrá alterar la seguridad de la red a su voluntad.

DMZ o Red de Zona Perimetral



Un nuevo tipo de firewall surge del hecho de que una táctica muy popular para separar las redes internas de las grandes corporaciones de los entornos hostiles de la Red, es construir una “red de encaminamiento” (es decir, formada sólo por routers) por la cual debe pasar todo el tráfico de entrada y de salida de la red corporativa.



Las grandes instalaciones disponen de este tipo de red para separar el tráfico local del tráfico con las redes de área amplia.

¿Por qué el nombre de DMZ? DMZ es el acrónimo de “Zona desmilitarizada” (demilitarized zone) y tiene el mismo propósito que en las zonas de conflicto geográfico: es una “tierra de nadie” entre dos partes hostiles que deben coexistir en proximidad una de la otra.

La existencia de una zona perimetral implica una seguridad multinivel. Primero, existen, al menos, dos routers implicados en la protección de la red interna. Un router actúa como el gateway a Internet y el otro como el gateway a la red interna.

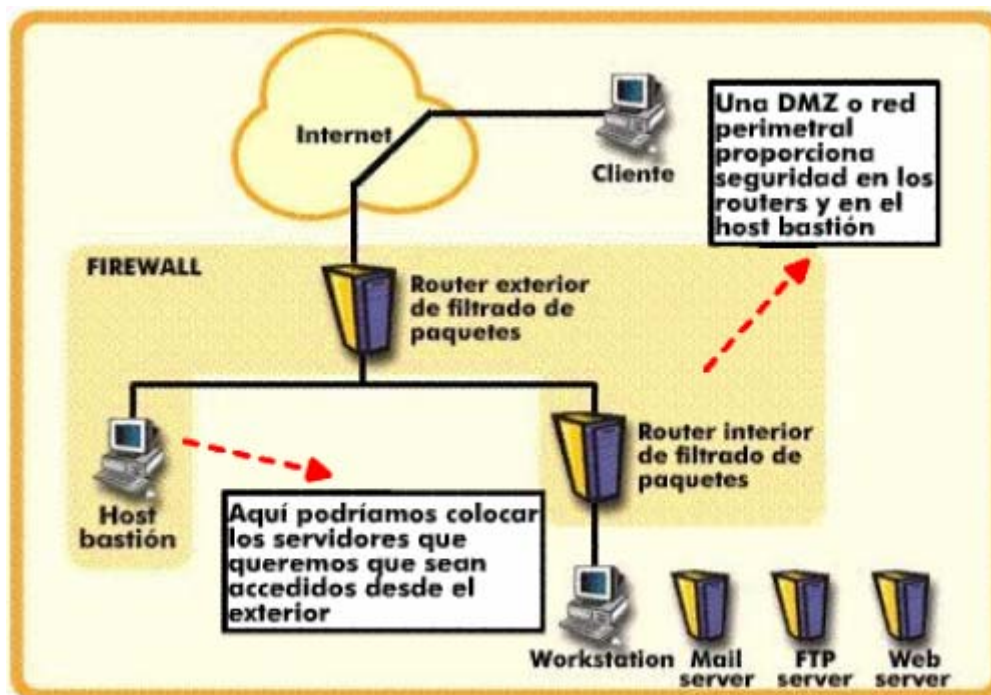


La red que comparten los dos routers no tiene ningún otro dispositivo conectado a ella que no sea un router o un equipo de confianza (utilizado como un host bastión).

El segundo nivel de seguridad inherente a la arquitectura DMZ aparece cuando se produce alguna brecha en la red perimetral; los intrusos sólo podrán detectar los paquetes que circulen por dicha red, nada más. Una vez que se ha obtenido el acceso a la red perimetral, todavía hay que vencer la resistencia del router interno de dicha red, lo cual puede desmotivar a los posibles atacantes.

Además, una solución VPN para la red interior es muy probable que implique la utilización de paquetes encriptados, lo cual complicará más la entrada de los intrusos.

Configuración de una DMZ estándar



En la construcción de una DMZ estándar, los controles más complejos se sitúan en el router interno, el cual separa la red interior de la red perimetral y de la red exterior.

Por ejemplo, muchas veces se coloca un NAT (Network Address Translator) en el router interno para complicar la localización y el secuestro de las direcciones IP de los equipos conectados a la red interna.

NAT proporciona seguridad porque traduce, de forma dinámica, direcciones no enrutables a direcciones reales de Internet.

De este modo, no se hace nada fácil el intercambio con los equipos internos.

La máxima seguridad que podemos obtener con una DMZ se consigue prohibiendo todo el tráfico desde la red interna hacia el router exterior, y prohibiendo todo el tráfico desde Internet hacia la red interior.

En esencia, esto equivale a forzar un proceso de dos pasos en todo el tráfico.

Los clientes en Internet sólo pueden dialogar con máquinas que se encuentren en la DMZ, y los clientes que se encuentran en lo más profundo de la red interior no verán directamente a Internet; necesitarán la intermediación de un host bastión en la DMZ.

¿Qué tipo de hosts bastión colocaremos en la DMZ?

Generalmente, servidores de aplicaciones, que puedan ser accedidos tanto desde Internet como desde la red interna.

Como ya hemos mencionado, la intención en muchas de las acciones que tomamos enfocadas a la seguridad, tratan de dificultar al máximo el "fisgoneo" de los intrusos.



Cuanto más difícil les resulte, antes se desanimarán. Lograr un sistema totalmente inviolable puede que no sea posible.

Servidores proxy



Por último, veremos un tipo de firewall sobre el que puede existir alguna polémica sobre si se trata de un nuevo tipo o una variedad de un tipo anterior. ¿A qué nos referimos?



Un proxy actúa de manera muy parecida a un host bastión, y en algunos libros sobre firewalling, no existe ninguna distinción entre estas dos figuras. Utilizamos el término “host bastión” para referirnos a un ordenador que actúa como etapa intermedia en la información que se transmite desde/hacia Internet.

El término “servidor proxy” tiene para nosotros un significado diferente; lo utilizamos para referirnos a un tipo especial de host bastión que ejecuta un software específico que enmascara una máquina interna en otra externa. Un proxy puede verse como un dispositivo de filtrado de paquetes, pero a nivel de aplicación. En el ejemplo siguiente vamos a contrastar un host bastión típico con un servidor proxy típico.



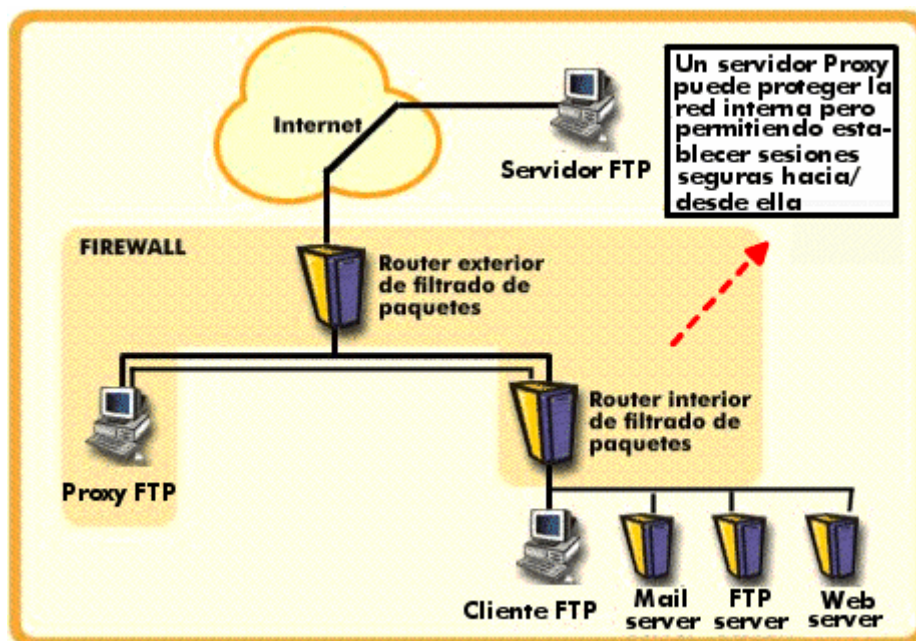
Un buen ejemplo de una aplicación de host bastión es el correo electrónico. El host bastión actúa como “punto de envío” para el correo entrante desde Internet.

Desde aquí, el bastión puede reenviar el correo hacia un servidor de correo interior (actuando como lo que se conoce por router de correo), o puede almacenar el correo, esperando que el cliente lo lea como un cliente de correo POP (Post Office Protocol) (actuando como servidor de correo electrónico). Se pueden construir muchos más ejemplos de firewall siguiendo esta filosofía.



Sin embargo, un servicio proxy es más un punto de control “en tránsito” que una etapa de almacenamiento de información. El proxy pretende ser el extremo de una conexión, pero que protege al verdadero transmisor o receptor de la información del tráfico no deseado.

Ejemplo de un servidor proxy



Un buen ejemplo de proxy lo tenemos en la presente ilustración. Aquí utilizamos una sesión FTP a través de un firewall con un proxy.

El cliente de la red interna establece una conexión con el proxy de la red perimetral, el cual arrancará por él el servicio FTP contra un servidor en Internet. El proxy suplanta al cliente interno y trabaja con el servidor remoto en su lugar.

Cualquier cliente FTP externo que quisiese acceder a un servidor interno FTP, lo haría a través del proxy, actuando este último, en beneficio del cliente remoto, contra el servidor local FTP.

En este caso, la utilización del proxy dentro de la DMZ nos asegura que los accesos al servidor local FTP van a ser seguros.